

QE for Display [RX] サンプルプログラム

要旨

本アプリケーションノートでは、ルネサス製 RX マイコンに対応する統合開発環境 e² studio 用のプラグイ ンである"QE for Display [RX]"と連携するサンプルアプリケーションについて説明します。"QE for Display [RX]"は、表示機器を搭載した組み込みシステム開発において、表示制御をグラフィカルな I/F で サポートするツールです。"QE for Display [RX]"を使用してシステム開発を行うためには、RX ファミリ に搭載されているグラフィク LCD コントローラ(以下、GLCDC)を初期化するためのプログラムが必要で す。本アプリケーションノートでは、GLCDC を初期化するためのベースとなるサンプルプログラムを提供 します。

対象デバイス

- ・RX65N グループ、RX651 グループ ROM 容量:1.5MB ~ 2MB
- ・RX72N グループ
- ・RX72M グループ
- ・RX66N グループ

本アプリケーションノートは、

- Renesas Starter Kit+ for RX72N、
- Renesas Envision KIT RPBRX72N
- Renesas Starter Kit+ for RX65N-2MB
- Renesas Envision KIT RPBRX65N

向けのサンプルプログラムを用意しています。

その他のデバイス、ボードに適用する場合は、「10 ユーザ環境に適用するには」を参照してください。

目次

1. 概要 1.1	QE for Display [RX]を使った開発のフロー	4 5
2. 動作	環境	6
3. 関連	ドキュメント	10
4. サン	プルプロジェクトの構成	11
5. サン 5.1 5.2 5.3 5.4 5.5	プルプロジェクトの実行手順 QE for Display [RX]のインストール プロジェクトのインポート プロジェクトのビルド デバッガ接続とプログラムの実行 QE for Display [RX]によるリアルタイム調整	12 13 14 16 17 19
6. //— 6.1 6.2	ドウェア説明 ハードウェア構成 端子機能	20 20 20
7. ソフ 7.1 7.2 7.2.1 7.2.2 7.2.3 7.2.4 7.3 7.4 7.5 7.6 7.7 7.8 7.8 7.8.1 7.8.2	トウェア説明 動作概要 GLCDC の動作設定詳細 パラメータ設定(QE) パラメータ設定(ユーザ) GLCDC の端子設定 ボードの端子設定 GLCDC FIT モジュールのパラメータと QE for Display [RX]が出力するヘッダとの対応 周辺機能 メモリ メモリ メモリ メモリ メモリ タモリマップ 使用割り込み一覧 本サンプルの使用に際して 変数一覧	23 24 25 25 25 25 26 34 35 36 36 37
8. QE f 8.1 8.2 8.3 8.4 8.5 8.6 8.7 8.8	or Display [RX]の使用方法 QE for Display[RX]の起動 LCD パネルデータの設定 制御信号の出力設定 LCD パネル制御信号タイミングの調整 制御信号出力/タイミング調整結果の反映 イメージダウンロード機能 各種画質の調整 画質調整用ヘッダファイルの生成	38 39 39 41 43 44 46 48
9. LCD 9.1 9.2 9.3 9.4	パネルデータの設定詳細 登録名称の記入 表示方式の選択 制御タイミングの入力 作成したディスプレイデータの編集	49 49 50 51 52
10. ユー 10.1 10.1	ザ環境に適用するには 仕様確認 1 パネルクロック	53 54 54

RENESAS

10.1.2	端子割り付け(ビットエンディアン)とピクセル配列	54
10.1.3	その他の制御端子	55
10.2 プロ	コジェクトの作成	56
10.2.1	Smart Configurator の選択	56
10.2.2	セクションの設定	57
10.3 Sm	art Configurator の設定	58
10.3.1	クロックの設定	58
10.3.2	GLCDC FIT モジュールの追加	58
10.3.3	GLCDC で使用する端子の設定	58
10.4 QE	for Display [RX]による調整(初期設定)	58
10.5 プロ	コグラムの作成	59
10.5.1	サンプルプログラムのコピー	59
10.5.2	プログラムの変更	59
10.6 実行	テから調整終了まで	60

1. 概要

GLCDC は、図 1-1 に示すように複数のブロックで構成されているため、表示確認を行うだけでも GLCDC の仕様を理解しいくつもの設定を行う必要があります。そこで、本サンプルプログラムと QE for Display [RX]を使用する事で GLCDC の仕様を理解する必要なく短期間で表示機器の接続確認ができる環境 を準備しました。QE for Display [RX]は、表示制御をグラフィカルな I/F でサポートするツールです。使用 する表示機器の情報を入力することで、表示制御に必要な情報を含んだヘッダファイルを出力します。その ヘッダファイルをもとにサンプルプログラムが GLCDC の設定を行います。

また、ツールにはリアルタイムでタイミングを調整する機能があり、使用する表示機器を接続したまま微調整を行ってから、ヘッダファイルを出力する事も可能です。

さらに、デバイスの設定をグラフィカルにサポートする Smart Configurator や、RX ファミリのドライバ/ ミドルウェアを提供する Firmware Integration Technology(以下、FIT)を使用し、QE for Display [RX]と連携することで、より簡単に表示制御を行うことが可能になります。本サンプルでは、QE for Display [RX] を 主体に、Smart Configurator および FIT が提供する "グラフィック LCD コントローラモジュール Firmware Integration Technology"(以下、GLCDC FIT モジュール)を使用しています。



以降に本サンプルについて記載します。

図 1-1 GLCDC のブロック構成

1.1 QE for Display [RX]を使った開発のフロー

QE for Display [RX]を使用したシステム開発のフローを図 1-2 に示します。



図 1-2 QE for Display [RX]を使用したシステム開発

2. 動作環境

本サンプルは、Renesas Starter Kit+ for RX72N(以下、RSK RX72N)、Renesas Envision KIT RPBRX72N(以下、Envision RX72N)、Renesas Starter Kit+ for RX65N-2MB(以下、RSK RX65N)と Renesas Envision KIT RPBRX65N(以下、Envision RX65N)で動作を確認しています。

各ボードの動作確認条件を以下示します。

表 2-1	動作確認条件	(RSK RX72N)
12 2-1	刧IF 唯心不 IT	

項目	内容
使用マイコン	R5F572NNDDBD (RX72N グループ)
動作周波数	• メインクロック: 24MHz
	● PLL: 240MHz (メインクロック 1 分周 10 逓倍)
	● システムクロック(ICLK): 240MHz (PLL 1 分周)
	● 周辺モジュールクロック A(PCLKA): 120MHz (PLL 2 分周)
	● 周辺モジュールクロック B(PCLKB): 60MHz (PLL 4 分周)
	● LCD パネルクロック(LCD_CLK): 10MHz (PLL 24 分周)
動作電圧	3.3V
統合開発環境	ルネサスエレクトロニクス製
	e ² studio Version 7.7.0
Cコンパイラ	ルネサスエレクトロニクス製
	C/C++ Compiler Package for RX Family V.3.01.00
	コンパイルオプション
	-lang = c99
iodefine.h のバージョン	Version 1.0
エンディアン	リトルエンディアン、ビッグエンディアン
動作モード	シングルチップモード
プロセッサモード	スーパバイザモード
サンプルプログラムのバージョン	Version 1.10
エミュレータ	E2 Lite
使用ボード	Renesas Starter Kit+ for RX72N(型名:RTK5572NNxxxxxxxx)

項目	内容
使用マイコン	R5F572NNHDFB (RX72N グループ)
動作周波数	• メインクロック: 16MHz
	● PLL: 240MHz (メインクロック 1 分周 15 逓倍)
	● システムクロック(ICLK): 240MHz (PLL 1 分周)
	● 周辺モジュールクロック A(PCLKA): 120MHz (PLL 2 分周)
	● 周辺モジュールクロック B(PCLKB): 60MHz (PLL 4 分周)
	● LCD パネルクロック(LCD_CLK): 10MHz (PLL 24 分周)
動作電圧	3.3V
統合開発環境	ルネサスエレクトロニクス製
	e ² studio Version 7.7.0
Cコンパイラ	ルネサスエレクトロニクス製
	C/C++ Compiler Package for RX Family V.3.01.00
	コンパイルオプション
	-lang = c99
iodefine.h のバージョン	Version 1.0
エンディアン	リトルエンディアン、ビッグエンディアン
動作モード	シングルチップモード
プロセッサモード	スーパバイザモード
サンプルプログラムのバージョン	Version 1.10
エミュレータ	E2OB (E2 オンボードエミュレータ)
使用ボード	Renesas Envision KIT RPBRX72N
	(製品型名: RTK5RX72N0CxxxxxBJ)
ボード設定	<sw1></sw1>
(ジャンパ / スイッチ)	Pin 1: don't care
	Pin 2: OFF
	(デバッガを使用する)
	<その他>
	デフォルト

表 2-2 動作確認条件(Envision RX72N)

表 2-3 動作確認条件(RSK RX65N)

項目	内容
使用マイコン	R5F565NEDDFC (RX65N グループ)
動作周波数	• メインクロック: 24MHz
	● PLL: 240MHz (メインクロック 1 分周 10 逓倍)
	● システムクロック(ICLK): 120MHz (PLL 2 分周)
	● 周辺モジュールクロック A(PCLKA): 120MHz (PLL 2 分周)
	● 周辺モジュールクロック B(PCLKB): 60MHz (PLL 4 分周)
	● LCD パネルクロック(LCD_CLK): 10MHz (PLL 24 分周)
動作電圧	3.3V
統合開発環境	ルネサスエレクトロニクス製
	e ² studio Version 7.7.0
Cコンパイラ	ルネサスエレクトロニクス製
	C/C++ Compiler Package for RX Family V.3.01.00
	コンパイルオプション
	-lang = c99
iodefine.h のバージョン	Version 2.3
エンディアン	リトルエンディアン、ビッグエンディアン
動作モード	シングルチップモード
プロセッサモード	スーパバイザモード
サンプルプログラムのバージョン	Version 1.10
エミュレータ	E2 Lite
使用ボード	Renesas Starter Kit+ for RX65N-2MB
	(製品型名: RTK50565Nxxxxxxxx)
ボード設定	<sw4></sw4>
(ジャンパ / スイッチ)	Pin 3:OFF
	Pin 4:ON
	(LCD を使用する)
	<その他>
	デフォルト

項目	内容
使用マイコン	R5F565NEDDFB (RX65N グループ)
動作周波数	• メインクロック: 12MHz
	● PLL: 240MHz (メインクロック 1 分周 20 逓倍)
	 システムクロック(ICLK): 120MHz (PLL 2 分周)
	● 周辺モジュールクロック A(PCLKA): 120MHz (PLL 2 分周)
	● 周辺モジュールクロック B(PCLKB): 60MHz (PLL 4 分周)
	● LCD パネルクロック(LCD_CLK): 10MHz (PLL 24 分周)
動作電圧	3.3V
統合開発環境	ルネサスエレクトロニクス製
	e ² studio Version 7.7.0
Cコンパイラ	ルネサスエレクトロニクス製
	C/C++ Compiler Package for RX Family V.3.01.00
	コンパイルオプション
	-lang = c99
iodefine.h のバージョン	Version 2.3
エンディアン	リトルエンディアン、ビッグエンディアン
動作モード	シングルチップモード
プロセッサモード	スーパバイザモード
サンプルプログラムのバージョン	Version 1.10
エミュレータ	E2OB (E2 オンボードエミュレータ)
使用ボード	Renesas Envision KIT RPBRX65N
	(製品型名: RTK5RX65N2CxxxxxBR)
ボード設定	<sw1></sw1>
(ジャンパ / スイッチ)	Pin 1: ON
	Pin 2: OFF
	(テパッカを使用する)
	<sw4></sw4>
	Pin 1: OFF
	PIII 2. don t care (デバッガを使用する)
	(ノハツリを使用する) ここの曲。
	<て (7)1世> ニ コ エ μ ト
	Pin 1: OFF Pin 2: don't care (デバッガを使用する) <その他>
	テフォルト

表 2-4 動作確認条件(Envision RX65N)

3. 関連ドキュメント

本サンプルに関連するドキュメントを以下に示します。併せて参照してください。

Firmware Integration Technology ユーザーズマニュアル (R01AN1833) RX ファミリ ボードサポートパッケージモジュール Firmware Integration Technology (R01AN1685) RX ファミリ グラフィック LCD コントローラモジュール Firmware Integration Technology (R01AN3609) Renesas e² studio スマート・コンフィグレータ ユーザーガイド (R20AN0451) RX65N グループ Renesas Starter Kit+ for RX65N-2MB ユーザーズマニュアル (R20UT3888) RX65N Group RX65N Envision Kit User's Manual (R01UH0761) RX65N グループ、RX651 グループ ユーザーズマニュアル ハードウェア編 (R01UH0590) RX72N グループ Renesas Starter Kit+ for RX72N ユーザーズマニュアル (R20UT4436) RX72N Group RX72N Envision Kit User's Manual (R20UT4788) RX72N グループ ユーザーズマニュアル ハードウェア編 (R01UH0824)

最新版がある場合、最新版に差し替えて使用してください。最新版はルネサスエレクトロニクスホームページで確認および入手してください。

4. サンプルプロジェクトの構成

本サンプルのプロジェクト構成を示します。なお、FIT モジュールや統合開発環境で自動生成されるファ イルの詳細は除きます。

サンプルプロジェクトはボードごとに4つ用意しています。プロジェクト構成自体は同じで、QE for Display[RX]を使い LCD に合わせて出力したヘッダファイルとマイコンに依存するファイルが異なります。 図 4-1 プロジェクトのフォルダ構成 では、代表して RSK RX72N プロジェクトのフォルダ構成を詳細に 示しています。

表 4-1 サンプルのプロジェクト

プロジェクト名	概要
QE_for_Display_sample_RX72N_RSK	RSK RX72N で動作するプロジェクト
QE_for_Display_sample_RX72N_Envision	Envision RX72N で動作するプロジェクト
QE_for_Display_sample_RX65N_RSK	RSK RX65N で動作するプロジェクト
QE_for_Display_sample_RX65N_Envision	Envision RX65N で動作するプロジェクト



図 4-1 プロジェクトのフォルダ構成

5. サンプルプロジェクトの実行手順

本章では本サンプルプロジェクトで QE for Display [RX]のリアルタイム調整を行うまでの実行手順を示し ます。QE for Display [RX]のリアルタイム調整機能を使用するには、GLCDC の初期化が完了している必要 があります。GLCDC の初期化はユーザプログラムで実行します。

本サンプルのプログラムでは GLCDC の初期化を glcdc_initialize 関数で行っています。glcdc_initialize 関数の実行が完了してから QE for Display [RX] によるリアルタイム調整を開始してください。 本サンプルの詳細については、7. ソフトウェア説明を参照してください。

なお、本プロジェクトを動作させる前に 2. 動作環境に示すジャンパ設定がある場合は、必ず設定してく ださい。

QE for Display [RX]の使用方法については、8. QE for Display [RX]の使用方法を参照してください。

事前準備

1. QE for Display [RX]のインストール

実行手順

- 2. プロジェクトのインポート
- 3. プロジェクトのビルド
- 4. デバッガ接続とプログラムの実行
- 5. QE for Display [RX]によるリアルタイム調整

5.1 QE for Display [RX]のインストール

まず初めに QE for Display [RX]を統合開発環境 e² studio にインストールします。インストールは下記の 手順で行います。

<u>インストール方法</u>

- 1. e² studio を起動する。
- 2. [ヘルプ(H)]→[新規ソフトウェアのインストール...]メニューを選択し、[インストール]ダイアログを開く。
- 3. [追加(A)...]ボタンを押下し、[リポジトリーを追加]ダイアログを開く。
- [アーカイブ(A)...]ボタンを押下し、開いたファイル選択ダイアログで、インストール用 zip ファ イル(RenesasQE_display_package_V130.zip)を選択し、[開く(O)]ボタンを押下する。
- 5. [リポジトリーを追加]ダイアログで、[OK]ボタンを押下する。
- 6. [必要なソフトウェアを見つけるために、インストール中に更新サイト全てに接続(C)]チェック ボックスを外します。
- 7. [インストール]ダイアログに、表示された[Renesas QE for Display[RX]チェックボックスをチェックし、[次へ(N)>]ボタンを押下する。
- 8. インストール対象が [Renesas QE for Display[RX]] となっていることを確認し、[次へ(N)>]ボタンを押下する。
- 9. ライセンスを確認した後、[使用条件の条項に同意します(A)]ラジオ・ボタンを選択し、[終了(F)] ボタンを押下する。
- 10. 信頼する証明書の選択ダイアログが表示された場合、表示された証明書をチェックした後、[OK] ボタンを押下してインストールを継続する。
- 11. e² studio の再起動を促されるので再起動を行う。
- 尚、アンインストールは下記の手順で行います。

<u>アンインストール方法</u>

- 1. e² studio を起動する。
- [ヘルプ(H)]→[e2studio について(A)]メニューを選択し、[インストール詳細(i)]ボタンを押下して、 [e² studio のインストール詳細]ダイアログを開く。
- 3. [インストールされたソフトウェア]タブに表示されている[Renesas QE for Display[RX]]を選択し、[アンインストール(U)…]ボタンを押下して、[アンインストール]ダイアログを開く。
- 4. 表示された内容を確認し、[終了(F)]ボタンを押下する。
- 5. e² studio の再起動を促されるので再起動を行う。

- 5.2 プロジェクトのインポート
 - 1. [ファイル(F)]をクリックしてください。
 - 2. [インポート(I)]をクリックしてください。

	新規(N)	Alt+シフト+N >	
	ファイルを開く(.)		
2	ファイル・システムからプロジェクトを開く		[ファイル(F)]クリックしてください
	閉じる(C)	Ctrl+W	
	すべて閉じる(L)	Ctrl+シフト+W	
	保管(S)	Ctrl+S	
	別名保存(A)		
3	すべて保管(E)	Ctrl+シフト+S	
	前回保管した状態に戻す(T)		
	移動(V)		
	名前を変更(M)	F2	
1	更新(F)	F5	
	行区切り文字の変換(D)	>	
5	印刷(P)	Ctrl+P	[インポート(I)]クリックしてください
3	インポート(I)		
2	エクスポート(O)		
	プロパティ(R)	Alt+Enter	
	ワークスペースの切り替え(W)	>	
	再開		
	終了/出口(X)		

- 3. [一般]の [既存プロジェクトをワークスペースへ]をクリックしてください。
- 4. [次へ(N)>]をクリックしてください。

R ≥ a	
^	
	■ [既存プロジェクトをワークスペースへ]をクリックしてください
·	━ [次へ(N)>]をクリックしてください

- 5. [ルート・ディレクトリーの選択(T):]のコンボボックスに本サンプルのプロジェクトが格納されている フォルダを指定してください。
- 6. [終了(F)]をクリックしてください。

e ² インポート	– 🗆 ×	
プロジェクトのインボート 既存の Eclipse プロジェクトを検索するディレクトリーを選択します。		本サンプルのプロジェクトが格納されているフォルダを
●ルート・ディレクトリーの選択(圧: C¥download¥workspace)	✓ 参照(<u>R</u>) 参照(R)	指定してください。
QE_for_Display_sample_RX65N_Envision(C:¥download¥worksp	すべて選択(<u>S</u>)	
✓ QE_tor_Display_sample_RX65N_RSK(C:¥download¥workspace¥ ✓ QE_for_Display_sample_RX72N_Envision(C:¥download¥workspace¥	選択をすべて解除(<u>D</u>	0
QE_for_Display_sample_RX72N_RSK(C:¥download¥workspace¥	更新(<u>E</u>)	
< オブション ネストしたプロジェクトを検索(H) フプロジェクトをワークスペースにコピー(C) ワークスペースに既に存在するプロジェクトを隔す(j)		プロジェクトをワークスペースにコピーしたい場合、 チェックを入れてください。
ワーキング・セット		
□ ワーキング・セットにプロジェクトを追加(I)	新規(<u>W</u>)	
ワーキング・セット(<u>ロ</u>): ~	選択(<u>E</u>)	
		[終了(F)]をクリックしてください
(?) < 戻る(B) 次へ(N) > 終了(E)	キャンセル	

5.3 プロジェクトのビルド

以下の手順に従い、プロジェクトをビルドしてロードモジュールを作成してください。

- 1. ビルドするプロジェクト(例:QE_for_Display_sample_RX72N_RSK HardwareDebug)をクリック してください。
- 2. [Build]をクリックしてください。

e ² qe_for_display_rx_sample - e ² studio ファイル(F) 編集(E) ソース(S) リファクタリング(T) ナ	ビゲート(N) 検索(A) プロジェクト(P) Renesas Views 実行(R) ウインド I QE_for_Display_sample_RX72N_RSF 💊 🌞 📄 😁 👻 🗐 🐚 🛛 😵
 	C [*] QE_for_Display_sample_RX72N_RSK HardwareDebug C [*] QE_for_Display_sample_RX65I ビルドするプロジェクトをクリックしてください C [*] QE_for_Display_sample_RX72N_Envision HardwareDebug C [*] QE_for_Display_sample_RX65N_Envision HardwareDebug

3. 「コンソール」パネルに「'Build complete.'」と表示されたらビルド完了です。

🔝 問題 📃 コンソール 🛛 🍓 スマート・ブラウザー CDT ビルド・コンソール [QE_for_Display_sample_RX72N_RSK] 'Finished building target:' C:\Renesas\e2_studio_v770_20191031\Utilities\\ccrx\renesas_cc_cc Loading input file QE_for_Display_sample_RX72N_RSK.abs Parsing the ELF input file..... 37 segments required LMA fixes Converting the DWARF information.... Constructing the output ELF image.... Saving the ELF output file QE_for_Display_sample_RX72N_RSK.x 'Build complete.' 16:02:21 Build Finished. 0 errors, 0 warnings. (took 15s.573ms)

5.4 デバッガ接続とプログラムの実行

- 1. デバッグするプロジェクト(例:QE_for_Display_sample_RX72N_RSK HardwareDebug)をクリック してください。
- 2. [Launch in ' \vec{r} in' \vec{r} (D)' mode]をクリックしてください。

e ² qe_for_display_rx_sample - e ² studio
ファイル(F) 編集(E) ソース(S) リファクタリング(T) ナビゲート(N) 検索(A) プロジェクト(P) Renesas Views 実行(R) ウィンド
🐔 🝺 📄 🐐 デバッグ(B) 🗸 📴 QE_for_Display_sample_RX72N_RSI へ 🌼 🗄 マ 🔚 🐚 🖉
🗃 🕶 📽 🕶 🚱 🖛 🚱 🖛 🥵 🖛 🎥 🛷 📴 🧭 📴 QE_for_Display_sample_RX72N_RSK HardwareDebug
▶ プロジェクト・エクスプローラー 🛛 🖳 🖳 📴 QE_for_Display_sample_R》 デバッグする プロジェクトをクリックしてください
CE for Dirday cample RVSENI Envision CE for Dirday cample RVSENI Envision
[Launch in 'デバッグ(D)' mode]をクリックしてください y sample RX65N Envision HardwareDebug
> 🔁 QE_for_Display_sample_RX72N_Envision
> 😂 QE_for_Display_sample_RX72N_RSK [Hardwa

3. 以下のメッセージが表示されたら、[はい(Y)]をクリックしてください。

┏² パースペクティブ切り替えの確認		\times		
この種類の起動は、中断時に デバッグ パースペクティブが開くように構成されています。				
このデバッグ・パースペクティブは、アプリケーション・デバッグをサポートするために設計されています。これには、デバッグ・スタック、変数、およびブレークポイント管理を表示するビューが組み込まれています。				
このパースペクティブを開きますか?				
[はい(Y)]クリ	ックします			
	はい(Y) いいえ(N)			

4. ロードモジュールのダウンロードが完了すると、[デバッグ]パースペクティブが開きます。

● ● A = 」 月 = 前 = む 中 = ○				9199-7922 B GC++	なデバッグ
\$ 71/57 TI © [Cl. Let, Diplay, sample, RCXN, NSK HardnardDebrg, Benessa GOB HardnardDebrg, Benessa V © Cl. Let, Diplay, sample, RCXN, NSK HardnardDebrg, Benessa GOB HardnardDebrg, Benessa V © Cl. Let, Diplay, sample, RCXN, NSK HardnardDebrg, Benessa GOB HardnardDebrg, Benessa The rest of the sector is and an access double flag. (7.8.2) © Remets GOB invert (Hot)	□ 00-変数 23 0 ₀ プレークボ・ 名前	ント 語レジスター 100 モジュール 空 式 タイプ	₽® イベントポイント] IC 値	8 egisters 🔬 🥶 🖻 📑 🖻	
	٢			> <	>
<pre>Startest the startest sta</pre>	mments above */ ther than register variable to the U stack (if change e on these nichtalined e _DNITSCT() is executed.	· · / · · · · · · · · · · · · · · · · ·	◆ ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	PRV PRV JAN PRV PRV PRV PRV PRV PRV PRV PRV PRV PRV	i dd
2 37/-5 11 (2) 737 (2) 782 (2) 288 (4) 777-12 (2) Geograph Cannon (1) ステレーサイスアン(名誉 9 (2) (2) (2) (2) (2) (2) (2) (2) (2) (2)	(QE)		U A K		• • • • •

5. ツールバーの[再開]をクリックしてください。プログラムが実行され、main 関数の先頭でブレークします。

ノパ1ル(上) 福朱(上)	ソース(<u>S</u>) リファクタリング(T) ナビゲート(<u>N</u>) 検索(<u>A</u>) プロジェクト(<u>P</u>) Renesas <u>V</u> iews	実行(R) ウィンドウ(W) ヘルプ(H)	
s 🐐 🔳	☆ デパッグ(B) ∨ C QE_for_Display_sample_RX72N_RSI ∨ 🌼	} - 🔒 🕼 🗞 - 🗞 - 🛍 😂 🗙	D 🛛 🖉 🕅 🎝 🗇 🖄 🖬 🤿 🕅
🍅 🍅 🔗 🕶 🌛	½ ▼ { ▼ 🏷 🗘 ▼ 🖒 ▼		
☆ デバッグ ☆	‰ i+ é	闷 ▽ 🖵 🖪 (x)= 変数 5? 🔍 ブルークポ	イント 💷 レジスター 🛋 モジュール 😪 式 🍺 イベ
V C QE_for_Displ	ay_sample_RX72N_RSK HardwareDebug [Renesas GDB Hardware Debugging]	クリックしてください	N
v w are_no_b v w are for a f	ppn/jamp/control (Tricorce) ジグナル: SIGTRAP:Trace/breakpoint trap) werON_Reset PC() at resetprg.c:188 0xffc019bd > rx-force-isa=v3 -rx-force-double-fpu (7.8.2) JDB server (Host)	*4 B)	917
			- 8
💼 resetprg.c 🔀			
i resetprg.c ∷ 188 ffc019bd	□ R_BSP_POR_FUNCTION(R_BSP_STARTUP_FUNCTION)		^
c resetprg.c ☆ 188 ffc019bd 189 190 191	<pre> R_BSP_POR_FUNCTION(R_BSP_STARTUP_FUNCTION) { /* Stack pointers are setup prior to calling this function /* Stack pointers are setup prior to calling this function /* Stack pointers are setup prior to calling this function /* Stack pointers are setup prior to calling this function /* Stack pointers are setup prior to calling this function /* Stack pointers are setup prior to calling this function /* Stack pointers are setup prior to calling this function /* Stack pointers are setup prior to calling this function /* Stack pointers are setup prior to calling this function /* Stack pointers are setup prior to calling this function /* Stack pointers are setup prior to calling this function /* Stack pointers are setup prior to calling this function /* Stack pointers are setup prior to calling this function /* Stack pointers are setup prior to calling this function /* Stack pointers are setup prior to calling this function /* Stack pointers are setup prior to calling this function /* Stack pointers are setup prior to calling this function /* Stack pointers are setup prior to calling this function /* Stack pointers are setup prior to calling the setup prior to calling</pre>	ı - see comments above */	Â.
resetprg.c ☆ 188 ffc019bd 189 190 191 192 193 194	 R BSP POR_FUNCTION(R_BSP_STARTUP_FUNCTION) /* Stack pointers are setup prior to calling this function /* You can use auto variables in this function but such var	n - see comments above */ riables other than register variabl ? I stack to the U stack (if change	es :). */

6. main 関数の先頭でブレークした後に、もう一度ツールバーの[再開]をクリックしてください。

< * I	○ 人会 ジングジング(i) シビン (i) マンジング(i) シビンジング(i) シビンジング(i) マンジング(i) マビンジング(i) シビンジング(i) シビンジング(i) マンジング(i) マンジンジング(i) マンジング(i) マンジンジンジング(i) マンジング(i) マンジンジンジンジンジング(i) マンジンジンジンジンジンジンジンジンジンジンジンジンジンジンジンジンジンジンジ	L_RSF ∨ 🄅 🕴 📬 ▼ 🔡 🦷) · · · · · · · · · · · · · · · · · · ·	🕨 II 🔳 N R. 🕞
🥭 🅭 🛷 🔹 🛛				_
☆ デバッグ ≥3		🔆 i i i 🖗 🗢	クリックしてください	ト 胡辞 レジスター 🛋 モミ
V 🤌 Thr	:ad #1 1 (single core) [core: 0] (Suspended : ブレークポイント) main() at main.c:147 0xffc0076a		24 BU	917
🔎 rx-elf-	ıdb -rx-force-isa=v3 -rx-force-double-fpu (7.8.2) s GDB server (Host)		٢	
Renesa Renesa	db -rx-force-isa=v3 -rx-force-double-fpu (7.8.2) s GDB server (Host) ⓒ main.c 었		<	

7. 表示機器の設定が正しく行われた場合、LCD パネルに以下の画面が表示されます。



5.5 QE for Display [RX]によるリアルタイム調整

1. LCD パネルに画面が表示されたら、QE for Display [RX] を起動し、リアルタイム調整を開始してください。





6. ハードウェア説明

6.1 ハードウェア構成

本サンプルで使用する LCD パネルを表 6-1 に示します。

表 6-1	サンプルで使用する LCD パネル
-------	-------------------

ボード	LCD パネル製品情報				
RSK RX72N	メーカ: Newhaven Display 社製				
RSK RX65N	型番: NHD-4.3-480272EF-ATXL#-CTP				
	画面サイズ:480×272				
	同期信号: VS、HS、DE (3 信号)				
	タッチコントローラ搭載(本サンプルでは使用しません)				
Envision RX72N	メーカ:EastRising 社製				
Envision RX65N	型番: ER-TFT043-3				
	画面サイズ:480×272				
	同期信号: VS、HS、DE (3 信号)				
	タッチコントローラ搭載(本サンプルでは使用しません)				

6.2 端子機能

各 RSK および Envision で使用する端子と機能を以下に示します。使用する製品に合わせて端子を選択してください。なお、端子機能は Smart Configurator を使用して設定できます。

表 6-2 使用端子と機能(RSK RX72N)

接続デバイス	端子名	入出力	内容
NHD-4.3-	P14/LCD_CLK-B	出力	パネルクロック出力
480272EF-ATXL#-	P13/LCD_TCON 0-B	出力	同期信号(VSYNC)出力
CIP	PJ2/LCD_TCON 2-B	出力	同期信号(HSYNC)出力
	PB1/LCD_TCON 3-B	出力	同期信号(DE)出力
	PC5/LCD_DATA 0-B	出力	LCD 信号出力 R[3]
	P82/LCD_DATA 1-B	出力	LCD 信号出力 R[4]
	P81/LCD_DATA 2-B	出力	LCD 信号出力 R[5]
	P80/LCD_DATA 3-B	出力	LCD 信号出力 R[6]
	PC4/LCD_DATA 4-B	出力	LCD 信号出力 R[7]
	P55/LCD_DATA 5-B	出力	LCD 信号出力 G[2]
	P54/LCD_DATA 6-B	出力	LCD 信号出力 G[3]
	P11/LCD_DATA 7-B	出力	LCD 信号出力 G[4]
	P83/LCD_DATA 8-B	出力	LCD 信号出力 G[5]
	PC7/LCD_DATA 9-B	出力	LCD 信号出力 G[6]
	PC6/LCD_DATA 10-B	出力	LCD 信号出力 G[7]
	PJ0/LCD_DATA 11-B	出力	LCD 信号出力 B[3]
	P85/LCD_DATA 12-B	出力	LCD 信号出力 B[4]
	P84/LCD_DATA 13-B	出力	LCD 信号出力 B[5]
	P57/LCD_DATA 14-B	出力	LCD 信号出力 B[6]

QE for Display [RX] サンプルプログラム

接続デバイス	端子名	入出力	内容
	P56/LCD_DATA 15-B	出力	LCD 信号出力 B[7]
	P27/汎用入出力ポート	出力	バックライト(プログラムで制御)
	PK4/汎用入出力ポート	出力	パネルリセット(プログラムで制御)

表 6-3 使用端子と機能(Envision RX72N)

接続デバイス	端子名	入出力	内容
ER-TFT043-3	PB5/LCD_CLK-B	出力	パネルクロック出力
	PB4/LCD_TCON 0-B	出力	同期信号(VSYNC)出力
	PB2/LCD_TCON 2-B	出力	同期信号(HSYNC)出力
	PB1/LCD_TCON 3-B	出力	同期信号(DE)出力
	PB0/LCD_DATA 0-B	出力	LCD 信号出力 B[3]
	PA7/LCD_DATA 1-B	出力	LCD 信号出力 B[4]
	PA6/LCD_DATA 2-B	出力	LCD 信号出力 B[5]
	PA5/LCD_DATA 3-B	出力	LCD 信号出力 B[6]
	PA4/LCD_DATA 4-B	出力	LCD 信号出力 B[7]
	PA3/LCD_DATA 5-B	出力	LCD 信号出力 G[2]
	PA2/LCD_DATA 6-B	出力	LCD 信号出力 G[3]
	PA1/LCD_DATA 7-B	出力	LCD 信号出力 G[4]
	PA0/LCD_DATA 8-B	出力	LCD 信号出力 G[5]
	PE7/LCD_DATA 9-B	出力	LCD 信号出力 G[6]
	PE6/LCD_DATA 10-B	出力	LCD 信号出力 G[7]
	PE5/LCD_DATA 11-B	出力	LCD 信号出力 R[3]
	PE4/LCD_DATA 12-B	出力	LCD 信号出力 R[4]
	PE3/LCD_DATA 13-B	出力	LCD 信号出力 R[5]
	PE2/LCD_DATA 14-B	出力	LCD 信号出力 R[6]
	PE1/LCD_DATA 15-B	出力	LCD 信号出力 R[7]
	P67/汎用入出カポート	出力	バックライト(プログラムで制御)
	PB3/汎用入出カポート	出力	パネルリセット(プログラムで制御)

表 6-4 使用端子と機能(RSK RX65N)

接続デバイス	端子名	入出力	内容
NHD-4.3-	PB5/LCD_CLK-B	出力	パネルクロック出力
480272EF-ATXL#-	PB4/LCD_TCON 0-B	出力	同期信号(VSYNC)出力
CIP	PB2/LCD_TCON 2-B	出力	同期信号(HSYNC)出力
	PB1/LCD_TCON 3-B	出力	同期信号(DE)出力
	PB0/LCD_DATA 0-B	出力	LCD 信号出力 R[3]
	PA7/LCD_DATA 1-B	出力	LCD 信号出力 R[4]
	PA6/LCD_DATA 2-B	出力	LCD 信号出力 R[5]
	PA5/LCD_DATA 3-B	出力	LCD 信号出力 R[6]
	PA4/LCD_DATA 4-B	出力	LCD 信号出力 R[7]
	PA3/LCD_DATA 5-B	出力	LCD 信号出力 G[2]
	PA2/LCD_DATA 6-B	出力	LCD 信号出力 G[3]
	PA1/LCD_DATA 7-B	出力	LCD 信号出力 G[4]

QE for Display [RX] サンプルプログラム

接続デバイス	端子名	入出力	内容
	PA0/LCD_DATA 8-B	出力	LCD 信号出力 G[5]
	PE7/LCD_DATA 9-B	出力	LCD 信号出力 G[6]
	PE6/LCD_DATA 10-B	出力	LCD 信号出力 G[7]
	PE5/LCD_DATA 11-B	出力	LCD 信号出力 B[3]
	PE4/LCD_DATA 12-B	出力	LCD 信号出力 B[4]
	PE3/LCD_DATA 13-B	出力	LCD 信号出力 B[5]
	PE2/LCD_DATA 14-B	出力	LCD 信号出力 B[6]
	PE1/LCD_DATA 15-B	出力	LCD 信号出力 B[7]
	PB7/汎用入出カポート	出力	バックライト(プログラムで制御)
	P97/汎用入出カポート	出力	パネルリセット(プログラムで制御)

表 6-5 使用端子と機能(Envision RX65N)

接続デバイス	端子名	入出力	内容
ER-TFT043-3	PB5/LCD_CLK-B	出力	パネルクロック出力
	PB4/LCD_TCON 0-B	出力	同期信号(VSYNC)出力
	PB2/LCD_TCON 2-B	出力	同期信号(HSYNC)出力
	PB1/LCD_TCON 3-B	出力	同期信号(DE)出力
	PB0/LCD_DATA 0-B	出力	LCD 信号出力 B[3]
	PA7/LCD_DATA 1-B	出力	LCD 信号出力 B[4]
	PA6/LCD_DATA 2-B	出力	LCD 信号出力 B[5]
	PA5/LCD_DATA 3-B	出力	LCD 信号出力 B[6]
	PA4/LCD_DATA 4-B	出力	LCD 信号出力 B[7]
	PA3/LCD_DATA 5-B	出力	LCD 信号出力 G[2]
	PA2/LCD_DATA 6-B	出力	LCD 信号出力 G[3]
	PA1/LCD_DATA 7-B	出力	LCD 信号出力 G[4]
	PA0/LCD_DATA 8-B	出力	LCD 信号出力 G[5]
	PE7/LCD_DATA 9-B	出力	LCD 信号出力 G[6]
	PE6/LCD_DATA 10-B	出力	LCD 信号出力 G[7]
	PE5/LCD_DATA 11-B	出力	LCD 信号出力 R[3]
	PE4/LCD_DATA 12-B	出力	LCD 信号出力 R[4]
	PE3/LCD_DATA 13-B	出力	LCD 信号出力 R[5]
	PE2/LCD_DATA 14-B	出力	LCD 信号出力 R[6]
	PE1/LCD_DATA 15-B	出力	LCD 信号出力 R[7]
	P66/汎用入出カポート	出力	バックライト(プログラムで制御)
	P63/汎用入出カポート	出力	パネルリセット(プログラムで制御)

7. ソフトウェア説明

7.1 動作概要

本サンプルは、クロックや割り込みなどの CPU の初期化を行い、QE for Display [RX]から出力されたヘッダファイルをもとに GLCDC の動作設定を行います。



図 7-1 サンプルプログラムの動作概要

7.2 GLCDC の動作設定詳細

図 7-2 に GLCDC の動作設定詳細を示します。GLCDC の設定は、GLCDC FIT モジュールを通して行いま す。GLCDC FIT モジュールの初期化関数である R_GLCDC_Open 関数の引数には、GLCDC FIT モジュー ルが用意している glcdc_cfg_t 構造体の変数を設定します。この構造体変数に、画面表示のためのいくつも のパラメータを設定します。

このパラメータに設定する情報は、以下に分類されます。

a. GLCDC が出力する同期信号や RGB 信号に関する設定	\rightarrow	QE for Display [RX]	を使って編集
b. 入力データに対する出力補正の設定	\rightarrow	QE for Display [RX]	を使って編集
c. GLCDC に入力する画像データに関する設定	\rightarrow	ユーザがプログラムを	を直接編集
d. 割り込み関連の設定	\rightarrow	ユーザがプログラムを	を直接編集

QE for Display [RX]では、主に LCD パネルの仕様に依存する a. と b. のパラメータ設定をサポートして います。残りの c. と d. に関しては、使用する画像の形式やシステムに応じてユーザがプログラムを直接 編集する必要があります。ユーザは構造体変数の c. d. に該当するパラメータに対して値を指定します。

パラメータの設定が完了してから、GLCDC FIT モジュールの初期化関数(R_GLCDC_Open)にパラ メータを渡します。この後、端子の設定を行います。

GLCDC の端子設定 ボードの端子設定 → Smart Configurator を使って編集 → ユーザがプログラムを直接編集

端子設定後、GLCDC FIT モジュールの制御関数(R_GLCDC_Control)で画面表示を開始します。



図 7-2 GLCDC の動作設定詳細

7.2.1 パラメータ設定(QE)

ここで呼び出す qe_for_display_parameter_set 関数は、本サンプルで用意した関数です。GLCDC FIT モ ジュールに設定するパラメータの内、QE for Display [RX]が出力する define 定義を使うパラメータをまとめ ています。

7.2.2 パラメータ設定(ユーザ)

GLCDC FIT モジュールに設定するパラメータの内、QE for Display [RX]で調整しないパラメータを設定します。使用する画像の形式やシステムに応じてユーザがプログラムを直接編集する必要があります。ユーザ は構造体変数の c. d. に該当するパラメータに対して値を設定します。本サンプルにおける設定値は、7.3 GLCDC FIT モジュールのパラメータと QE for Display [RX]が出力するヘッダとの対応を参照してください。

7.2.3 GLCDC の端子設定

6.2 端子機能で GLCDC の端子設定は Smart Configurator で設定できると説明しました。ここで呼び出す R_GLCDC_PinSet 関数は、Smart Configurator が生成した r_glcdc_rx_pinset.c ファイル内に実装されてい ます。

Smart Configurator で GLCDC の端子設定ができる様子を図 7-3 に示します。LCD パネルの仕様と接続に応じて、各端子機能を使用する/使用しないの選択と、各端子機能に割り当てるポート番号を指定します。 端子設定の方法については、Smart Configurator のマニュアルを参照してください。



図 7-3 GLCDC の端子設定画面 (Smart Configurator)

7.2.4 ボードの端子設定

各ボードには、GLCDC が制御する端子の他に LCD パネルのバックライトやリセット端子を制御するためのポートが接続されています。

ここで呼び出す board_port_setting 関数では、LCD パネルのバックライトやリセット端子の制御を行います。これら端子の制御はユーザがプログラムを直接編集する必要があります。

7.3 GLCDC FIT モジュールのパラメータと QE for Display [RX]が出力するヘッダとの 対応

GLCDC FIT モジュールに設定するパラメータ(R_GLCDC_Open 関数の引数に設定する glcdc_cfg_t 構造 体メンバ)と、QE for Display [RX]が出力する define 定義の対応を表 7-1 に示します。サンプルの設定値の 列で、"←"の記載になっているものは、QE for Display [RX] が出力する define 定義で設定しています。 "←"以外の記載になっているものは、ユーザが設定します。パラメータの詳細は、GLCDC FIT モジュール のマニュアルを参照してください。

表 7-1 GLCDC FIT モジュールのパラメータと QE for Display [RX] が出力する定義の対応

概略	構造体メンバ	QE for Display [RX]が 出力する define 定義	サンプルの 設定値	設定内容
 水平バック ポーチ	output.htiming.	LCD_CH0_W_HBP		水平バックポーチ期間
・ ・ 水平アサー ト幅	output.htiming. sync_width	LCD_CH0_W_HSYNC	←	水平同期信号期間
垂直バック ポーチ	output.vtiming. back_porch	LCD_CH0_W_VBP	←	垂直バックポーチ期間
垂直アサー ト幅	output.vtiming. sync_width	LCD_CH0_W_VSYNC	←	垂直同期信号期間
水平有効表 示幅	output.htiming. display_cyc	LCD_CH0_DISP_HW	←	水平有効表示期間
垂直有効表 示幅	output.vtiming. display_cyc	LCD_CH0_DISP_VW	←	垂直有効表示期間
水平フロン トポーチ	output.htiming. front_porch	LCD_CH0_W_HFP	←	水平フロントポーチ期間
垂直フロン トポーチ	output.vtiming. front_porch	LCD_CH0_W_VFP	←	垂直フロントポーチ期間
コールバッ ク関数への ポインタ	p_callback	-	glcdc_callba ck	割り込み要因発生時にポインタが 示すアドレスのコールバック関数 を実行します
クロック ソース	output.clksrc	-	GLCDC_CL K_SRC_INT ERNAL	PLL クロックを使用
クロックの 分周比	output. clock_div_ratio	-	GLCDC_PA NEL_CLK_D IVISOR_24	LCD_CLK の分周比の設定
出力データ フォーマッ ト	output.format	LCD_CH0_OUT_FORMA T	<i>←</i>	出力データフォーマット
TCON、 DATA の出 力位相制御	output. sync_edge	LCD_CH0_OUT_EDGE	←	LCD_CLK の立ち上がり/立ち下が りに同期して出力
水平同期信 号(HSYNC) の出力端子	output. tcon_hsync	LCD_CH0_TCON_PIN_ HSYNC	←	HSYNC の出力に使用する TCON を選択
水平同期信 号(HSYNC) の極性	output. hsync_polarity	LCD_CH0_TCON_POL_ HSYNC	←	極性をロー/ハイアクティブに設定

機能 構造体スシハ CE for Lisplay [FX,]か サンルの ま皮内容 建画問期程 (VSYNC) output. fton_wsync LCD_CH0_TCON_PIN_V SYNC - VSYNC の出力に使用する TCON を選択 2010月37 output. ton_wsync LCD_CH0_TCON_POL VSYNC - 優先をロー/ハイアクティブに設定 7年94 output. ton_det LCD_CH0_TCON_PIN_ VSYNC - 優任をロー/ハイアクティブに設定 7年7 output. ton_de LCD_CH0_TCON_PIN_ DE - 優任をロー/ハイアクティブに設定 7年7 output. ton_de LCD_CH0_TCON_PIN_ DE - BE の出力に使用する TCON を選 分 7年7 output. ton_de LCD_CH0_TCON_PIN_ DE - BE の出力に使用する TCON を選 分 7年7 output. ton_de LCD_CH0_TCON_PIN_ DE - BE の出力に使用する TCON を選 分 7年7 output. ton_det LCD_CH0_TCON_PIN_ DE - BE の出力に使用する TCON を選 分 7年7 output. ton_det LCD_CH0_TCON_PIN_ DE - BE の出力に使用する TCON を認 分 - 7年7 output. ton_det LCD_CH0_TCON_PIN_ DE - - - - 7年7 output. ton_det CCC TSCO - - -<					
出力する define 定義 設定値 exit [minited] output. toon_vsync LCD_CHO_TCON_PIN_U SYNC ~ VSYNC output. 2SYNC VSYNC output. 2SYNC attem parts output. vsync_polarity LCD_CHO_TCON_POL_ VSYNC ~ ~ Rtete n=///1479574712%2 7-94 (OE)004 output. toon_de LCD_CHO_TCON_PIN_ DE ~ DE DE output. 0E DE output. DE 7-94 (At=_n/n E output. toon_de LCD_CHO_TCON_POL_ DE ~ Mtete n=///1479574712%2 7-94 (At=_n/n E output. toon_de LCD_CHO_TCON_POL_ DE ~ Mtete n=///1479574712%2 7-94 (At=_n/n E output.bg_color. byte.g LCD_CHO_TCON_POL_ DE ~ Mtete n=///1479574712%2 7 output.bg_color. byte.g - OXCC 常景色o R 値の設定 ? 7 D	概略	構造体メンバ	QE for Display [RX]	サンフルの	設定内容
普廣開期程 (VSYNC) output. con_vsync (vsn.cplaitly) CDC CH0 TCON_PIN_V SYNC			出力する define 定義	設定値	
QFUCSINC 動出の違子 (SYNC) con_vsync (SYNC) SYNC 差選択 重直開期信 QVDSINC) output. vsync_polarity LCD_CHO_TCON_POL_ VSYNC - 種性をロー///イアクティブに設定 データイ A - フル信 Q(DE)の出 Na# output. ton_de LCD_CHO_TCON_PIN_ DE - DE DE の出力に使用する TCON を選 R プータイ A - フル信 Q(DE)の出 the anable pola rity output. con_de LCD_CHO_TCON_POL_ DE - Mete D - /// 47 クラティ ブに設定 R プータイ A - フル信 Q(DE)の超 the anable pola rity output. con_de LCD_CHO_TCON_POL_ DE - 歴 他 E ロ - /// 47 クラティ ブに設定 R P R 色 O G 値の設定 学業色 O Uutput. byte. output. byte. LCD_CHO_TCON_POL_ DE - Wete D - /// 47 クラティ ブに設定 P R 色 O G 値の設定 プロム メ つ T - ス output.bg_color. byte.b - OXCC P R 色 O R 値の設定 P R O R 値 O B 値の設定 プレム メ い T - T - マ > F input.format - OXCC P R 色 O B 値の設定 D P - A // 9 T - マ > F を選択 プレム メ い T - マ > F input.format - OXCC P R 色 O B 値の設定 D P - A // 9 T - マ > F & P // 9 R & D	垂直同期信	output.	LCD_CH0_TCON_PIN_V	\leftarrow	VSYNC の出力に使用する TCON
の出力朝子 一 一 一 一 豊富同期信 (VSYNC) output vsync_polatity LCD_CH0_TCON_POL VSYNC - 橋桂をロー///イアクティブに設定 プーライ ネーブル信 (DE)の出 toon_de LCD_CH0_TCON_PIN_ DE - DE の出力に使用する TCON を選 R プーライ ネーブル信 (DE)の出 output toon_de CD_CH0_TCON_PIN_ DE - DE の出力に使用する TCON を選 プーライ ネーブル信 (DE)の出 output toon_de CD_CH0_TCON_POL DE - DE の出力に使用する TCON を選 プーライ ネーブル信 (DE)の出 output to_color. byle. CD_CH0_TCON_POL DE - DE DE 開景色 G output tog_color. byle. CD_CH0_TCON_POL DE - - Mette D=//// CO TO L-A (1y) 270 input tog_color. byle. CD_CH0_TCON_POL DE - 0xCC 常景色の G 値の設定 TO L-A (1y) 270 input tog_color. byle. - - 0xCC アラフィック 2.: CCC IN FGRMAT 1 BBTS RGB カラフフィーマットを選択 TV L-A (1y) 270 input.p_base - - - - - TV L-A (1y) 270 input.p_base - - - - - - - - </td <td>号(VSYNC)</td> <td>tcon_vsync</td> <td>SYNC</td> <td></td> <td>を選択</td>	号(VSYNC)	tcon_vsync	SYNC		を選択
重要同期信	の出力端子	_ ,			
Bale Name (VSYNC) Output. vsync_polarity CDC_IND_ION_POL_ VSYNC Description Description <thdescription< th=""> Description</thdescription<>	垂直同期信	output		<u></u>	極性をロー/ハイアクティブに設定
NULL Production VSTNC VSTNC Description Descriprot is description Description <td>모(\/SVNC)</td> <td>vovno polority</td> <td></td> <td></td> <td></td>	모(\/SVNC)	vovno polority			
Under F - 5 4 A - 7 Jufa G(DE)004 Output. toon_de LCD_CH0_TCON_PIN_ DE 一 DE の出力に使用する TCON を選 形 7 - 7 Jufa G(DE)004 output. data_enable_pola nty LCD_CH0_TCON_POL_ DE - - 器性をローバメイアクティブに設定 7 - 7 Jufa G(DE)004 output. data_enable_pola nty LCD_CH0_TCON_POL_ DE - - 器性をローバメイアクティブに設定 7 - 7 Jufa G(DE)044 output.bg_color. byte.g - 0xCC 常景色の G 値の設定 7 - 7 Jufa G(DE)045 output.bg_color. byte.g - 0xCC 常景色の G 値の設定 7 - 7 Jufa G(DE)04 output.bg_color. byte.g - 0xCC 常景色の G 値の設定 7 - 7 Jufa G(DE)04 input.format - 0xCC 常景色の G 値の設定 7 - 7 Jufa G(DE)04 input.format - 0xCC ドス - 7 - 7 Jufa G(DE)04 - - 0xCC ドス - - 7 - 7 Jufa G(DE)04 input.format - - 0xCC ドス - - - 7 - 7 - 7 input.format - - - - - - -	与(VSTNC)	vsync_polanty	VSTINC		
$\overline{\tau} - \overline{\gamma} / \overline{\alpha}$ byte,LCD_CH0_TCON_PIN_ DE $-$ DEDE MIDIC@HT 5 TCON 5 2 R $7 - \overline{\gamma} / \overline{\alpha}$ $4 - \overline{\gamma} / \overline{\alpha}$ $4 - \overline{\gamma} / \overline{\alpha}$ output. data_enable_pola hyte,LCD_CH0_TCON_POL_ DE $ \overline{\alpha} \mathbb{R} \pm \alpha - 1/n 4 \overline{\gamma} / \overline{\gamma} - \overline{\gamma} - \overline{\gamma} / \overline{\alpha} \mathbb{R} \pm \alpha - 1/n 4 \overline{\gamma} / \overline{\gamma} - \overline{\gamma} / \overline{\alpha} \mathbb{R} \pm \alpha - 1/n 4 \overline{\gamma} / \overline{\gamma} - \overline{\gamma} / \overline{\alpha} \mathbb{R} \pm \alpha - 1/n 4 \overline{\gamma} / \overline{\gamma} - \overline{\gamma} / \overline{\alpha} \mathbb{R} \pm \alpha - 1/n 4 \overline{\gamma} / \overline{\gamma} - \overline{\gamma} / \overline{\alpha} \mathbb{R} \pm \alpha - 1/n 4 \overline{\gamma} / \overline{\gamma} - \overline{\gamma} / \overline{\alpha} \mathbb{R} \pm \alpha - 1/n 4 \overline{\gamma} / \overline{\gamma} - \overline{\gamma} / \overline{\alpha} \mathbb{R} \pm \alpha - 1/n 4 \overline{\gamma} / \overline{\gamma} - \overline{\gamma} / \overline{\alpha} \mathbb{R} \pm \alpha - 1/n 4 \overline{\gamma} / \overline{\gamma} - \overline{\gamma} / \overline{\alpha} \mathbb{R} \pm \alpha - 1/n 4 \overline{\gamma} / \overline{\gamma} - \overline{\gamma} / \overline{\alpha} \mathbb{R} \pm \alpha - 1/n 4 \overline{\gamma} / \overline{\gamma} - \overline{\gamma} / \overline{\alpha} \mathbb{R} \pm \alpha - 1/n 4 \overline{\gamma} / \overline{\gamma} - \overline{\gamma} / \overline{\alpha} \mathbb{R} \pm \alpha - 1/n 4 \overline{\gamma} / \overline{\gamma} - \overline{\gamma} / \overline{\alpha} \mathbb{R} \pm \alpha - 1/n 4 \overline{\gamma} / \overline{\gamma} - \overline{\gamma} / \overline{\alpha} \mathbb{R} \pm \alpha - 1/n 4 \overline{\gamma} / \overline{\gamma} - \overline{\gamma} / \overline{\alpha} \mathbb{R} \pm \alpha - 1/n 4 \overline{\gamma} / \overline{\gamma} - \overline{\gamma} / \overline{\alpha} \mathbb{R} \pm \alpha - 1/n 4 \overline{\gamma} / \overline{\gamma} - \overline{\gamma} / \overline{\alpha} \mathbb{R} \pm \alpha - 1/n 4 \overline{\gamma} / \overline{\gamma} - \overline{\gamma} / \overline{\alpha} \mathbb{R} \pm \alpha - 1/n 4 \overline{\gamma} / \overline{\gamma} / \overline{\alpha} / \overline{\alpha} / \overline{\alpha} + \frac{1}{n 4 \overline{\alpha} - 1/n 4 \overline{\alpha} / \overline{\alpha} / \overline{\alpha} / \overline{\alpha} + \frac{1}{n 4 \overline{\alpha} - 1/n 4 \overline{\alpha} / \overline{\alpha} / \overline{\alpha} - 1/n 4 \overline{\alpha} / \overline{\alpha} / \overline{\alpha} / \overline{\alpha} + \frac{1}{n 4 \overline{\alpha} - 1/n 4 \overline{\alpha} / \overline{\alpha} / \overline{\alpha} - \frac{1}{n 4 \overline{\alpha} - 1/n 4 \overline{\alpha} / \overline{\alpha} / \overline{\alpha} - \frac{1}{n 4 \overline{\alpha} - 1/n 4 \overline{\alpha} / \overline{\alpha} / \overline{\alpha} - \frac{1}{n 4 \overline{\alpha} - 1/n 4 \overline{\alpha} / \overline{\alpha} / \overline{\alpha} - \frac{1}{n 4 \overline{\alpha} - 1/n 4 \overline{\alpha} / \overline{\alpha} / \overline{\alpha} - \frac{1}{n 4 \overline{\alpha} - 1/n 4 \overline{\alpha} / \overline{\alpha} / \overline{\alpha} - \frac{1}{n 4 \overline{\alpha} - 1/n 4 \overline{\alpha} / \overline{\alpha} / \overline{\alpha} - \frac{1}{n 4 \overline{\alpha} - 1/n 4 \overline{\alpha} / \overline{\alpha} / \overline{\alpha} - \frac{1}{n 4 \overline{\alpha} - 1/n 4 \overline{\alpha} / \overline{\alpha} / \overline{\alpha} - \frac{1}{n 4 \overline{\alpha} / \overline{\alpha} / \overline{\alpha} - \frac{1}{n 4 \overline{\alpha} - 1/n 4 \overline{\alpha} / \overline{\alpha} / \overline{\alpha} - \frac{1}{n 4 \overline{\alpha} / \overline{\alpha} / \overline{\alpha} - \frac{1}{n 4 \overline{\alpha} - 1/n 4 \overline{\alpha} / \overline{\alpha} / \overline{\alpha} - \frac{1}{n 4 \overline{\alpha} / \overline{\alpha} / \overline{\alpha} - \frac{1}{n 4 \overline{\alpha} / \overline{\alpha} / \overline{\alpha} / \overline{\alpha} - \frac{1}{n 4 \overline{\alpha} - 1/n 4 \overline{\alpha} / \overline{\alpha} / \overline{\alpha} - \frac{1}{n 4 \overline{\alpha} / \overline{\alpha} / \overline{\alpha} - \frac{1}{n 4 \overline{\alpha} / \overline{\alpha} / \overline{\alpha} / \overline{\alpha} - \frac{1}{n 4 \overline{\alpha} / \overline{\alpha} / \overline{\alpha} - \frac{1}{n 4 \overline{\alpha} / \overline{\alpha}$	の極性				
ホーフル信 9(DE)の加 オープル信 9(DE)の極 性 DE DE 駅 オーブル信 9(DE)の極 性 output. data_enable_pola riv LCD_CH0_TCON_POL DE ー 極性をロー///イアクティブに設定 育素色 R値 9(DE)の極 性 output.bg_color. byte_g - OxCC 背景色の G 値の設定 育素色 B値 0utput.bg_color. byte_g - OxCC 背景色の G 値の設定 プレーム /xッフ 7 の 画像フォー マット input.bg_color. byte_g - OxCC 背景色の B 値の設定 ブレーム /xッフ 7 の 画像フォー マット input.format - OxCC 背景色の B 値の設定 ブレーム /xッフ 7 の 音優 0.7 input.format - OxCC 片景色の B 値の設定 ブレーム/xッフ 7 の 先調 7 ドレ ス input.p_base - グラフ 7 ッ 2. フレーム/xッフ 7 の先頭 7 ドレス の設定 ブレーム/xッフ 7 の 先頭 7 ドレ ス input.bg_color. byte. - グラフ 7 ッ 2. フレーム/xッフ 7 の先頭 7 ドレス の設定 ブレ 4 input.bg_color. byte. - グラフ 7 ッ 2. ブレーム/xッフ 7 の先頭 7 ドレス の設定 ブレ 4 input.bg_color. byte. - グラフ 7 ッ 2. グラフ 7 ッ 2. バレス の設定 ブレ 4 input.bg_color. byte. - グラフ 7 ッ 2. グラフ 7 ッ 2. グラフ 7 ッ 2. グラフ 7 ッ 2. グラフ 7 ッ 2. <td>データイ</td> <td>output.</td> <td>LCD_CH0_TCON_PIN_</td> <td><i>←</i></td> <td>DEの出力に使用する TCON を選</td>	データイ	output.	LCD_CH0_TCON_PIN_	<i>←</i>	DEの出力に使用する TCON を選
号(DE)の出 カ端子 オーフル信 等(DE)の極 は、コーカル信 (DE)の極 性 output. data_enable_poin ity LCD_CH0_TCON_POL DE ー 価性をロー///イアクティブに設定 背景色の確 (byte, byte, 育景色の確 outputbg_color. byte, outputbg_color. byte, つutputbg_color. - OXCC 背景色のG 値の設定 背景色の確 (byte, byte, outputbg_color. byte, coutputbg_color. - OXCC 背景色のG 値の設定 プレーム パッフアの 画像クォー マット outputbg_color. byte, outputbg_color. - OXCC 背景色のG 値の設定 プレーム パッフアの た顔アトレス パッフィの クラフィッ ク1.*表設定 - グラフィッ ク1.*表設定 グラフィッ ク1.*表設定 カラーフォーマットを選択 パッフアの CICDC_IN FORMAT_1 6875-749 ク1.*表設E カラーフォーマットを選択 パッファの ク1.*表設E プレーム パッフアの た顔アトレス パッファの ク2. - グラフィッ ク1.*表設E フレーム パッフアの た顔え。 パッコーマックを無効が良い ファックを の設定 フレーム パッフアの た顔え。 パッコーマックを 加速 クラフィック フレーム パッフアの た顔え。 パッフ クロ クロ クロ クロ クロ クロ クロ クロ クロ クロ クロ クロ クロ	ネーブル信	tcon_de	DE		択
力減子 ○ ○ ○ デーライ ペーブル6 も(ata_enable_poin %(b) 0utput. (ata_enable_poin byte, LCD_CHO_TCON_POL DE ↓ ↓ 様性をロー///イアクティブに設定 開景色 R値 %(DE)の極 性 0utputbg_color. byte, 0utputbg_color. byte, 0xCC 背景色の G 値の設定 背景色 B値 0utputbg_color. byte,b - 0xCC 背景色の G 値の設定 ブレーム 0utputbg_color. byte,b - 0xCC 背景色の B 値の設定 ブレーム 0utputbg_color. 	号(DE)の出				
$\overline{\gamma} = 9.7$ $x = 7.0 \mbox{ data a. nable_polantyLCD_CH0_TCON_POL_DE\cdots\overline{W}\overline{W}\overline{W}\overline{P} \equiv 0.6\overline{W} = 0output.bg_color.byte.g-0xCC\overline{P} \equiv 0.6 \mbox{ de mBE}\overline{P} \equiv 0.6 \mbox{ de m}\overline{W} = 0output.bg_color.byte.g-0xCC\overline{P} \equiv 0.6 \mbox{ de mBE}\overline{T} = 0.6 \mbox{ de m}byte.goutput.bg_color.byte.g-0xCC\overline{P} \equiv 0.6 \mbox{ de mBE}\overline{T} = -2 \mbox{ de m}\overline{W} = 0output.bg_color.byte.g-0xCC\overline{P} \equiv 0.6 \mbox{ de mBE}\overline{T} = -2 \mbox{ de m}\overline{W} = 0output.bg_color.byte.g-0xCC\overline{P} \equiv 0.6 \mbox{ de mBE}\overline{T} = -2 \mbox{ de m}\overline{W} = 0output.bg_color.byte.g-\sqrt{2.7} \mbox{ de mBE}\overline{D} = 7.7 \mbox{ de mBE}\overline{T} = -2 \mbox{ de mBE}\overline{W} = 0input.format-\sqrt{2.7} \mbox{ de mBE}\overline{D} = 7.7 \mbox{ de mBE}\overline{D} = 7.7 \mbox{ de mBE}\overline{T} = -2 \mbox{ de mBE}\overline{D} = -2 \mbox{ de mBE}-\sqrt{2.7} \mbox{ de mBE}\overline{D} = 7.7 \mbox{ de mBE}\overline{D} = 7.7 \mbox{ de mBE}\overline{T} = -2 \mbox{ de mBE}\overline{D} = -2 \mbox{ de mBE}-\sqrt{2.7} \mbox{ de mBE}\overline{D} = 7.7 \mbox{ de mBE}\overline{D} = 7.7 \mbox{ de mBE}\overline{T} = -2 \mbox{ de mBE}-\sqrt{2.7} \mbox{ de mBE}\overline{D} = 7.7 \mbox{ de mBE}\overline{D} = 7.7 \mbox{ de mBE}\overline{T} = -2 \mbox{ de mBE}-\sqrt{2.7} \mbox{ de mBE}\overline{D} = 7.7 \mbox{ de mBE}\overline{D} = 7.7 \mbox{ de mBE}\overline{T} = -2 \mbox{ de mBE}-\sqrt$	力端子				
ネーブル信 号(CE)の種 性 data_enable_pola nty DE Image: Construction of the sector plane DE 背景色 R 値 byte.r outputbg_color. byte.g - 0xCC 背景色 O G 値の設定 背景色 G 値 byte.g outputbg_color. byte.g - 0xCC 背景色 O G 値の設定 プレーム バッフ 7 0 画像 フォー マット input format - 0xCC 背景色 O G 値の設定 プレーム バッフ 7 0 画像 フォー マット input.format - 0xCC 背景色 O G 値の設定 プレーム バッフ 7 0 先頭 7 1/2 0 7 先頭 7 ドレ ス input.format - 0xCC if 景色 O G 値の設定 プレーム バッフ 7 0 先頭 7 ドレ ス input.format - 0xCC if 景色 O G 値の設定 if 景色 O G 値の設定 プレーム バッフ 7 0 先頭 7 ドレ ス input.format - グラフ 4 ク 7 * 7 * ク 2: 0xCC ガラーフォーマットを選択 グラフ 4 ク 2: 0xCC if 見 C O C の設定	データイ	output.	LCD CH0 TCON POL	<i>←</i>	極性をロー/ハイアクティブに設定
守(DE)の確 性 mty mty ご OXCC 背景色のR 値の設定 背景色の値 output.bg_color. byte.d - OXCC 背景色のG 値の設定 背景色の値 output.bg_color. byte.d - OXCC 背景色のG 値の設定 プレーム パッフアの 画像フォー マット output.bg_color. byte.b - OXCC 背景色のB 値の設定 プレーム パッフアの 大ジッフアの 大ジッフアの 大調アドレス パッフアの 大調アドレス - グラフィッ ク 1:未設定 カラーフォーマットを選択 プレーム パッフアの 大調アドレス - グラフィッ ク 1:未設定 - グラフィッ ク 2: CRAME_BU FBASE_ADD DR(NORORO 000) プレームパッフアの先頭アドレス の設定 プレーム パッフアの 大頭アドレ ス - - - グラフィッ ク 1:未設定 - プレーム パッフアの 大頭アドレ ス - - グラフィッ ク 1:未設定 - - プレーム パッフアの 大調アドレ ス - - グラフィッ ク 2: CRAME_BU FBASE_ADD DR(NORORO 000) - - - プリン マックを無効に設定 (global cdg 二 (global cdg	ネーブル信	data enable pola	DE		
1 (1) <	号(DF)の極	rity	22		
IL IL <thil< th=""> IL IL IL</thil<>	性	inty			
市泉色 R 値 Object Output Og_Color. ・ OutC 市泉色 R 値 O R 値 O R 座 皆景色 G 値 Output Dg_color. byte.g output Dg_color. byte.g oxCC 常景色 G 値 output Dg_color. byte.g oxCC 常景色 G 値 output Dg_color. byte.g oxCC 常景色 G 値 output Dg_color. byte.g vutput Dg_color. byte.g oxCC f景色 G 値 oyte.g hput.format f.g f.g	*모유 D は	and and the second and		000	北早舟のP佐の恐穴
Pistel G (a) Output.bg_color. byte.g Pite (a) Output.bg_color. byte.g Pite (a) OxCC 背景色の (a) 管景色の (a) Pite (a) TU-L (i) input.format - OXCC 背景色の (a) カラーフオーマットを選択 TV-L (i) input.format - OXCC FRAMAT_1 (B) カラーフオーマットを選択 TV-L (i) input.format - - グラフィッ ク 1:未設定 カラーフオーマットを選択 TV-L (i) input.format - - - - - TV-L (i) input.format - </td <td>月京巴下迴</td> <td>output.bg_color.</td> <td>-</td> <td>UXCC</td> <td>月京巴の下値の設定</td>	月京巴下迴	output.bg_color.	-	UXCC	月京巴の下値の設定
背景色 G 値 byte.g output.bg_color. byte.b - OxCC 背景色 O 値 @ 閲定 背景色 B 値 Output.bg_color. byte.b - OxCC 背景色 O 面 値 @ 閲定 フレーム バッファの 画像フォー マット input.format - グラフィッ ク 2: GLCC _IN, FORMAT_1 6BITS_ROB 565 カラーフォーマットを選択 フレーム バッファの 先頭アドレ ス input.p_base - グラフィッ ク 1: 未設定 フレームバッファの先頭アドレス の設定。 ブラフィック チョ頭アドレ ス input.p_base - グラフィッ ク 1: 未設定 フレームバッファの先頭アドレス の設定。 ブラ 1.2 の背景 音色 G 値 input.bg_color. byte.r - グラフィッ ク 1: NULL フレームバッファの先頭アドレス の設定 グラフィッ ク 1: NULL - グラフィッ ク 1: NULL フレームバッファの先頭アドレス の設定 - ブラ 1.2 の背景 音色 G 値 input.bg_color. byte.r - グラフィッ ク 1: NULL - ブラフィッ ク 1.2 の背 景色 G 値 input.bg_color. byte.b - - グラフィッ ク 1: 未設定 - ブラフィッ ク 1.2 の背 景色 G 値 input.bg_color. byte.b - - グラフィッ ク 1: 未設定 ブラフィック 1.2 の背景色の B 値 の設定 ブラフィック 1.2 の背 景色 G 値 input.bg_color. byte.b - - - - ブラフィック 1.2 00 行 景色 B 値 input.bg_color. byte.b		byte.r			
byle.g byle.g inquit.bg_color. byle.b inquit.bg_color. byle.b inquit.bg_color. byle.b inquit.format informat informat <thinformat< th=""> info</thinformat<>	背景色 G 値	output.bg_color.	-	0xCC	背景色の G 値の設定
背景色 6 値 byte.boutput bg_color. byte.bcolor. cOXCC常景色 0 B 値 0 設定フレーム バッフアの 画像フォー マットinput.format-グラフィッ 2. GLCDC_IN FORMAT_1 6BITS_RGBカラーフォーマットを選択 ク2. (ADDAC)フレーム バッファの 先頭 アドレ スパッファの 先頭 アドレ スパッファの 先見 PT PT スinput.p_base-グラフィッ 7.2. FRAME_BU DR(N00800)フレームバッファの先頭アドレス の設定。 アイックを集効に設定 (glcdc,cdg,tinput 以下の構造体) シィンの設定値は無視されます)グラフィッ ク 1.2.0 ㎡ 景色 8 値input.bg_color. byte.r-グラフィッ 2.2.0 ㎡ ソーク 2.2.0 ㎡ byte.g-グラフィッ ク 1.2.0 ㎡ 景色 6 値input.bg_color. byte.gグラフィッ 2.2.0 ㎡ 2.2.0 ㎡ 2.2.0 ㎡ 2.2.0 ㎡ 2.2.0 ㎡-グラフィッ ク 1.2.0 ㎡ 景色 6 値input.bg_color. byte.gグラフィッ ク 1.2.0 ㎡ 景色 7.2.0 ㎡ (glcdc,cdg,tinput U)グラフィッ ク 1.2.0 ㎡ 泉色 6 値input.bg_color. byte.gグラフィッ ク 1.2.0 ㎡ 泉色 6 値input.bg_color. byte.gグラフィッ ク 1.2.0 ㎡ 泉色 6 値input.bg_color. byte.gグラフィッ ク 1.2.0 ㎡ 泉色 6 値input.bg_color. yote.gグラフィッ ク 1.2.0 ㎡ 泉色 6 値input.bg_color. yote.g <t< td=""><td></td><td>byte.g</td><td></td><td></td><td></td></t<>		byte.g			
byte.b byte.	背景色 B 値	output.bg_color.	-	0xCC	背景色の B 値の設定
フレーム バッファの 画像フォー マット input.format ・ グラフィッ ク2: GLCDC_IN_ FORMAT_1 BBTS_RGB 565 カラーフォーマットを選択 フレーム バッファの 先頭アドレ ス input_p_base ・ グラフィッ ク 1:未設定 フレームバッファの先頭アドレス の設定。 ス input_p_base ・ グラフィッ ク 2: FRAME_BU PG(0x00800 000) フレームバッファの先頭アドレス の設定 グラフィッ ク 1:2 01常 景色 R 値 input.bg_color. byte.b ・ グラフィッ ク 1:2 01常 景色 B 値 ・ グラフィッ ク 1:未設定 グラフィック 1.2 01常景色の R 値 の設定 グラフィッ 7 1:2 01常 景色 B 値 input.bg_color. byte.b ・ ・ グラフィッ 7 1:来設定 グラフィッ 7 5: 0xCC グラフィック 1.2 01 の設定 グラフィッ 7 1:2 01常 景色 6 値 input.bg_color. byte.b ・ ・ グラフィッ 7 2: 0xCC グラフィック 1.2 01 の設定 グラフィッ 7 1:2 01常 景色 8 値 input.bg_color. byte.b ・ ・ グラフィッ 7 2: 0xCC グラフィック 1.2 01 の設定		byte.b			
パッフィの 画像フォー マット Inputioninal ク2: GLCDC_IN_ FORMAT_1 6BITS_RGB 565 バッフィの (GLCDC_IN_ FORMAT_1 6BITS_RGB 565 ブレーム パッファの 先頭アドレ ス inputp_base ・ グラフィッ ク1:未設定 フレームパッファの先頭アドレス の設定。 ブレーム パッファの 先頭アドレ ス inputp_base ・ グラフィッ ク1:未設定 フレームパッファの先頭アドレス の設定。 ブリレーム パッファの 先頭アドレ ス inputp_base ・ グラフィッ ク1:未設定 フレームパッファの先頭アドレス の設定。 グラフィック ク1:なの習 景色 R 値 inputbg_color. byte.g ・ グラフィッ ク1:未設定 グラフィック1:2 の背景色の G 値 の設定 グラフィッ ク1:未設定 ・ ・ グラフィッ ク1:未設定 グラフィッ ク1:2 の背景色の G 値 の設定 グラフィッ ク1:2 の背 景色 B 値 ・ ・ グラフィッ ク1:未設定 グラフィッ ク1:未設定 グラフィッ ク1:2 の背 景色 B 値 ・ ・ グラフィッ ク1:本設定 グラフィッ ク1:2 の背景色の B 値	フレーム	input format	_	グラフィッ	カラーフォーマットを選択
ボット・ マット マット input.p_base · · GLCDC_IN_ FORMAT_1 6BITS_RGB 565 · ブレーム パッファの 大頭アドレ ス input.p_base · · グラフィッ ク1:未設定 · · ブレーム パッファの 大頭アドレ ス input.p_base · · グラフィッ ク2: · · · グラフィッ ク · · · · グラフィッ ク1:RME_BU F_BASE_AD DR(0x00800 000) · · · グラフィッ ク1:2.07 景色R値 input.bg_color. byte.r · · · グラフィッ ク1:未設定 · · グラフィッ ク1:2.07 景色R値 input.bg_color. byte.b · · · · · グラフィッ ク1:2.07 ク1:2.07 2: 0xCC input.bg_color. byte.b · · · · · グラフィッ ク1:2.07 2: 0xCC · · · · · ·	バッファの	input.ionnat		クリント クリ・	
画は > 7 - マット マット GLC DC_IN_ FORMAT_1 GBITS_RGB 565 GLC DC_IN_ FORMAT_1 GBITS_RGB 565 フレーム パッファの 先頭アドレ ス input.p_base - グラフィッ 72: FRAME_BU FBASE_AD DR(0x00800 00) フレームパッファの先頭アドレス の設定。 ブレーム パッファの 先頭アドレ ス input.p_base - グラフィッ 72: FRAME_BU FBASE_AD DR(0x00800 00) フレームパッファの先頭アドレス の設定。 ブリーム パッファの 先頭アドレ ス input.p_base - グラフィッ 71: YO 71:NULL フレームパッファの先頭アドレス の設定。 ブリーム パッファの た頭アドレス の設定 - グラフィッ 72: ORCC フレームパッファの先頭アドレス の設定。 ブリーム パッファの ク2: ORCC - グラフィッ 71: LO DR(0x00800 00) - - グラフィッ 71: 大口 2: ORCC - - グラフィッ 71: 未設定 - - グラフィッ 71: 未設定 - - - - - グラフィッ 71: 大口 2: ORCC - - - - - グラフィッ 71: 大口 2: ORCC - - - - - - グラフィッ 71: 大口 2: ORCC - - - - - - - - - - - - - - - - <td>「一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一</td> <td></td> <td></td> <td></td> <td></td>	「一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一				
マット FORMAL_1 BBTS_RGB 565 FORMAL_1 BBTS_RGB 565 ブレーム バッファの 先頭アドレ ス input.p_base - グラフィッ ク1:来設定 FRAME_BU F_BASE_AD DR(0x0800 000) フレームバッファの先頭アドレス の設定。 ブレーム バッファの 先頭アドレ ス input.p_base - グラフィッ ク2: FRAME_BU F_BASE_AD DR(0x0800 000) NULL を設定すると対象グラ フィックを無効に設定 (glodc_cfg_tinput 以下の構造体 メンバの設定値は無視されます) グラフィッ ク1:20 ㎡ 景色 R値 input.bg_color. byte.g - グラフィッ ク1:未設定 グラフィック 7:来設定 グラフィック 7:マック 1,2 の背景色の G 値 の設定 グラフィッ ク1:20 ㎡ 景色 G 値 input.bg_color. byte.b - グラフィッ 7:来設定 グラフィッ 7:来設定 グラフィック 1,2 の背景色の G 値 の設定 グラフィッ 7:来設定 - グラフィッ 7:来設定 グラフィック 1,2 の背景色の G 値 の設定 グラフィッ 7:来設定				GLCDC_IN_	
グラフィッ パッファの 光頭アドレ ス input.p_base - グラフィッ ク1:未設定 ブレームパッファの先頭アドレス の設定。 ブリーム パッファの 光頭アドレ ス input.p_base - グラフィッ ク2: アレームパッファの先頭アドレス の設定。 ズッファ 光頭アドレ ス input.p_base - デースの ・グラフィッ ク2: アレームパッファの先頭アドレス の設定。 グライック バッファの ク2: FRAME_BU NULL を設定すると対象グラ フィックを無効に設定 (gloda_cfg_tinput 以下の構造体 メンパの設定値は無視されます) NULL を設定すると対象グラ フィックク ク1:NULL グラフィッ ク1.2 の背 景色 8 値 input.bg_color. byte.g - グラフィッ ク1:未設定 グラフィッ ク1:未設定 グラフィック 1.2 の背景色の 6 値 の設定 グラフィッ ク1.2 の背 景色 8 値 input.bg_color. byte.b - グラフィッ ク1:未設定 グラフィック 1.2 の背景色の 6 値 の設定 グラフィッ ク1:2 の常 input.bg_color. byte.b - グラフィッ ク1:未設定 グラフィック 1.2 の背景色の 8 値 の設定	マツト			FORMAT_1	
グラフィッ グリンム パッファの 先頭アドレス ス input.p_base ・ グラフィッ ク2: FRAME_BU F.BASE_AD DR(0x00800 000) フレームパッファの先頭アドレス の設定。 ブリーム パッファの 先頭アドレス ス input.p_base ・ グラフィッ ク2: FRAME_BU F.BASE_AD DR(0x00800 000) NULL を設定すると対象グラ フィックを無効に設定 (glcdc_cfg_tinput 以下の構造体 メンパの設定値は無視されます) グラフィッ ク1:2 の背 景色 R 値 input.bg_color. byte.g ・ グラフィッ ク1:AUL グラフィッ ク1:未設定 グラフィッ ク1:2 の背 景色 R 値 input.bg_color. byte.g ・ グラフィッ ク1:本設定 グラフィック1.2 の背景色の G 値 の設定 グラフィッ ク1:2 の背 景色 B 値 input.bg_color. byte.b ・ ・ グラフィッ ク1:本設定 グラフィック1.2 の背景色の G 値 の設定 グラフィッ ク1:2 の背 景色 B 値 input.bg_color. byte.b ・ ・ グラフィッ ク1:本設定 グラフィック1.2 の背景色の B 値 の設定				6BITS_RGB	
「レーム バッファの パッファの 先頭アドレ スinput.p_base「グラフィッ クラス・ クロームバッファの先頭アドレス の設定。ブレームバッファの先頭アドレス の設定。アレーム パッファの 先頭アドレ スinput.p_base・・グラフィッ クロームに クロームに クロームに アローム アローム アローム グラフィッ クロームNULL を設定すると対象グラ フィックを無効に設定 (gloda_cdg_tinput 以下の構造体 メンパの設定値は無視されます)グラフィッ ク 1,2 の背 景色 R 値input.bg_color. byte.g・グラフィッ クローム グラフィッ クロームグラフィッ グラフィッ クロームグラフィッ ク 1,2 の背 景色 G 値input.bg_color. byte.g・・グラフィッ クローム グラフィッ クロームグラフィッ グラフィッ クロームグラフィッ ク 1,2 の背 景色 B 値input.bg_color. byte.g・・グラフィッ クローム クロームグラフィッ グラフィッ クロームグラフィッ ク 1,2 の背 景色 B 値input.bg_color. byte.g・・・グラフィッ クローム グラフィッ クロームグラフィック 1,2 の背景色の B 値 の設定グラフィッ ク 1,2 の背 分したの・・・・グラフィック 1,2 の背景色の B 値 の設定				565	
ブレーム パッファの 光頭アドレ ス input.p_base ・ グラフィッ ク 1:未設定 ブレームパッファの先頭アドレス の設定。 大頭アドレ ス input.p_base ・ ダラフィッ ク 2: FRAME_BU FBASE_BD DR(0x00800 000) NULL を設定すると対象グラ フィックを無効に設定 (glode_cfg_tinput 以下の構造体 メンパの設定値は無視されます) グラフィッ ク 1:2 0 ㎡ 景色 R 値 input.bg_color. byte.g ・ グラフィッ ク 2: 0xCC グラフィック クラフィック 1.2 0 ㎡ クラフィック 1.2 0 ㎡ の設定 グラフィッ ク 1:未設定 ・ ・ グラフィッ ク 1.1 表設定 グラフィック 1.2 0 ㎡ の設定 グラフィッ ク 1:未設定 ・ ・ グラフィッ ク 1:未設定 グラフィック 1.2 0 ㎡ の設定 グラフィッ ク 1:未設定 ・ ・ グラフィッ ク 1:未設定 グラフィック 1.2 0 ㎡ の設定					
マレーム パッファの 先頭アドレ ス input.p_base - グラフィッ クラフィッ ク2: フレームパッファの先頭アドレス の設定。 大頭アドレ ス input.p_base - FRAME_BU F_BASE_AD DR(0x00800 NULL を設定すると対象グラ フィックを無効に設定 (glcdc_cfg_tinput 以下の構造体 yンパの設定値は無視されます) グラフィッ ク12.0 ㎡ 景色 R 値 input.bg_color. byte.r - グラフィッ 2 : 0xCC グラフィック クラフィック 1:未設定 グラフィック クラフィック 1:未設定 グラフィッ ク 1.2.0 ㎡ 景色 G 値 input.bg_color. byte.g - グラフィッ 2 : 0xCC グラフィック 1.2.0 ㎡ の設定 グラフィッ ク 1.2.0 ㎡ 景色 B 値 input.bg_color. byte.b - グラフィッ 2 : 0xCC グラフィック 1.2.0 ㎡ の設定 グラフィッ ク 1.2.0 ㎡ 景色 B 値 input.bg_color. byte.b - グラフィッ 2 : 0xCC グラフィック 1.2.0 ㎡ の設定				グラフィッ	
フレーム パッファの 先頭アドレ ス input.p_base - グラフィッ クシン (PRAME_BU PROX00800 00) フレームパッファの先頭アドレス の設定。 大頭アドレ ス - <t< td=""><td></td><td></td><td></td><td>ク 1:未設定</td><td></td></t<>				ク 1:未設定	
バッファの 先頭アドレ スイン ケームク 2: FRAME_BU FBASE_AD DR(0x00800 000)の設定。グラフィッ ク 1,2 0背 景色 R 値input.bg_color. byte.rグラフィッ ケラフィックを無効に設定 (glodc_cfg_tinput 以下の構造体 メンバの設定値は無視されます)グラフィッ ク 1,2 0背 景色 R 値input.bg_color. byte.rグラフィッ ク 2: 0xCCグラフィック ク 3フィック 1,2 0背景色の R 値 の設定グラフィッ ク 1,2 0背 景色 G 値input.bg_color. byte.g-グラフィッ ク 2: 0xCCグラフィック 1,2 0背景色の G 値 の設定グラフィッ ク 1,2 0背 景色 G 値input.bg_color. byte.g-グラフィッ ク 1:未設定グラフィッ ク 1:未設定グラフィッ ク 1,2 0背 景色 G 値input.bg_color. byte.g-グラフィッ ク 1:未設定グラフィッ ク 1:未設定グラフィッ ク 1,2 0背 景色 B 値input.bg_color. byte.b-グラフィッ ク 1:未設定グラフィック 1,2 の背景色の B 値 の設定	フレーム	input.p base	-	グラフィッ	フレームバッファの先頭アドレス
先頭アドレ スLLERAME_BU F_BASE_AD DR(0x00800 00)NULL を設定すると対象グラ フィックを無効に設定 (glcdc_cfg_tinput 以下の構造体 メンバの設定値は無視されます)グラフィッ ク 1,2 の背 景色 R 値input.bg_color. byte.rグラフィッ 2 0xCCグラフィック 1,2 の背景色の R 値 の設定グラフィッ ク 1,2 の背 景色 G 値input.bg_color. byte.g-グラフィッ 2 0xCCグラフィック 1,2 の背景色の G 値 の設定グラフィッ ク 1,2 の背 景色 G 値input.bg_color. byte.g-グラフィッ 2 0xCCグラフィック 1,2 の背景色の G 値 の設定グラフィッ ク 1,2 の背 景色 B 値input.bg_color. byte.b-グラフィッ 2 0xCCグラフィック 1,2 の背景色の G 値 の設定	バッファの	–		ク2:	の設定。
ス interface FIGHALOG FIGHALOG FIGHALOG FIGHALOG Character Caracter Caracter <t< td=""><td>先頭アドレ</td><td></td><td></td><td>FRAME BU</td><td>NULLを設定すると対象グラ</td></t<>	先頭アドレ			FRAME BU	NULLを設定すると対象グラ
イラフィッ ク 1,2 の背 景色 R 値 input.bg_color. byte.r - グラフィッ ク 1:NULL グラフィッ ク 1:NULL グラフィッ ク 1:NULL グラフィッ ク 1,2 の背 景色 R 値 input.bg_color. byte.r - グラフィッ ク 2: 0xCC グラフィッ ク 2: 0xCC グラフィック 1,2 の背景色の R 値 の設定 グラフィッ ク 1,2 の背 景色 G 値 input.bg_color. byte.g - グラフィッ ク 1:未設定 グラフィッ ク 2: 0xCC グラフィック 1,2 の背景色の G 値 の設定 グラフィッ ク 1,2 の背 景色 G 値 input.bg_color. byte.g - グラフィッ ク 1:未設定 グラフィック 1,2 の背景色の G 値 の設定 グラフィッ ク 1,2 の背 景色 B 値 input.bg_color. byte.b - グラフィッ ク 2: 0xCC グラフィック 1,2 の背景色の B 値 の設定	7			E BASE AD	コノックを無効に設定
グラフィッ ク 1,2 の背 景色 R 値 input.bg_color. byte.r - グラフィッ ク 1:NULL グラフィッ ク 2: 0xCC グラフィック 1,2 の背景色の R 値 の設定 グラフィッ ク 1,2 の背 景色 R 値 input.bg_color. byte.g - グラフィッ ク 1:未設定 グラフィッ ク 1:未設定 グラフィック 1,2 の背景色の R 値 の設定 グラフィッ ク 1,2 の背 景色 G 値 input.bg_color. byte.g - グラフィッ ク 1:未設定 グラフィッ ク 1:未設定 グラフィック 1,2 の背景色の G 値 の設定 グラフィッ ク 1,2 の背 景色 G 値 input.bg_color. byte.b - グラフィッ ク 2: 0xCC グラフィック 1,2 の背景色の B 値 の設定					
グラフィッ ク1,2 の背 景色 R 値 input.bg_color. byte.r - グラフィッ ク 1:NULL グラフィッ ク 1:NULL グラフィック 1,2 の背景色の R 値 の設定 グラフィッ ク 1,2 の背 景色 R 値 input.bg_color. byte.g - グラフィッ ク 1:未設定 グラフィッ ク 1:未設定 グラフィッ ク 1:未設定 グラフィッ ク 1,2 の背 景色 G 値 input.bg_color. byte.g - グラフィッ ク 2: 0xCC グラフィッ ク 2: 0xCC グラフィック 1,2 の背景色の G 値 の設定 グラフィッ ク 1,2 の背 景色 G 値 input.bg_color. byte.b - グラフィッ ク 2: 0xCC グラフィック 1,2 の背景色の B 値 の設定					(gicdc_cig_t.input 以下の構造体
グラフィッ ク1.2 の背 景色 R 値 input.bg_color. byte.r グラフィッ 2 0 CC グラフィッ ク 2 0 CC グラフィッ ク 2 0 CC グラフィック 1,2 の背景色の R 値 の設定 グラフィッ ク 1,2 の背 景色 G 値 input.bg_color. byte.g - グラフィッ ク 1:未設定 グラフィッ ク 2: 0 CC グラフィッ ク 1:未設定 グラフィッ ク 1,2 の背 景色 G 値 input.bg_color. byte.g - グラフィッ ク 2: 0 CC グラフィッ ク 2: 0 CC グラフィック ク 2: 0 CC グラフィッ ク 1:未設定 - グラフィッ ク 1:未設定 グラフィッ ク 1:未設定 グラフィック 1,2 の背景色の B 値 の設定				000)	メンハの設定値は無視されます)
グラフィッ ク 1,2 の背 景色 R 値input.bg_color. byte.r-グラフィッ ク 2: 0xCCグラフィッ ク 2: 0xCCグラフィック 1,2 の背景色の R 値 の設定グラフィッ グラフィッ ク 1,2 の背 景色 G 値input.bg_color. byte.gグラフィッ グラフィッ ク 1:未設定グラフィッ グラフィッ ク 2: 0xCCグラフィック 1,2 の背景色の G 値 の設定グラフィッ グラフィッ ク 1,2 の背 景色 G 値input.bg_color. byte.gグラフィッ グラフィッ ク 1:未設定グラフィック 1,2 の背景色の G 値 の設定グラフィッ グラフィッ ク 1,2 の背 景色 B 値input.bg_color. byte.bグラフィッ グラフィッ ク 1:未設定グラフィック 1,2 の背景色の B 値 の設定					
グラフィッ ク 1,2 の背 景色 R 値 input.bg_color. byte.r ・ グラフィッ ク 2: 0xCC グラフィッ ク 3: 7 2: 0xCC グラフィッ の設定 グラフィッ ク 1,2 の背 景色 G 値 input.bg_color. byte.g ・ ・ グラフィッ ク 1:未設定 グラフィッ ク 2: 0xCC グラフィッ ク 1:未設定 グラフィッ ク 1,2 の背 景色 G 値 input.bg_color. byte.g ・ ・ グラフィッ ク 1:未設定 グラフィッ ク 1:未設定 グラフィッ ク 1,2 の背 景色 B 値 input.bg_color. byte.b ・ ・ グラフィッ ク 1:未設定 グラフィック 1,2 の背景色の B 値 の設定				クラノイツ	
グラフィッ ク 1,2 の背 景色 R 値input.bg_color. byte.r-グラフィッ ク 2: 0xCCグラフィック 1,2 の背景色の R 値 の設定グラフィッ グラフィッ ク 1:未設定グラフィッ グラフィッ ク 1:未設定グラフィッ グラフィッ ク 1:未設定グラフィック 1,2 の背景色の G 値 の設定グラフィッ ク 1,2 の背 景色 G 値input.bg_color. byte.g-グラフィッ グラフィッ ク 1:未設定グラフィッ グラフィッ ク 1:未設定グラフィッ グラフィッ ク 1,2 の背 景色 G 値input.bg_color. byte.b-グラフィッ グラフィッ グラフィッ ク 1:未設定グラフィッ グラフィッ グラフィッ グラフィック 1,2 の背景色の B 値 の設定				ク1:NULL	
ク 1,2 の背 景色 R 値byte.rク 2: 0xCCの設定グラフィッ ク 1:未設定グラフィッ ク 1:未設定グラフィッ ク 1:未設定グラフィック 1,2 の背景色の G 値 の設定グラフィッ ク 1,2 の背 景色 G 値input.bg_color. byte.g-グラフィッ ク 2: 0xCCグラフィック 1,2 の背景色の G 値 の設定グラフィッ ク 1,2 の背 景色 B 値input.bg_color. byte.b-グラフィッ ク 1:未設定グラフィック 1,2 の背景色の B 値 の設定	グラフィッ	input.bg_color.	-	グラフィッ	グラフィック 1,2 の背景色の R 値
景色 R 値 ビー・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	ク 1,2 の背	byte.r		ク2:0xCC	の設定
グラフィッ ク1:未設定グラフィッ ク1:未設定グラフィック1,2の背景色のG値グラフィッ ク1,2の背 景色G値input.bg_color. byte.g-グラフィッ ク2: 0xCCグラフィッ の設定グラフィッ ク1:未設定グラフィッ ク1:未設定グラフィッ ク1:未設定グラフィッ ク3: 0xCCグラフィッ ク1,2の背 身た.b-グラフィッ ク3: 0xCCグラフィッ グラフィッ ク3: 0xCC	景色 R 値				
グラフィッ ク 1,2 の背 景色 G 値input.bg_color. byte.gク - <td></td> <td></td> <td></td> <td>グラフィッ</td> <td></td>				グラフィッ	
グラフィッ ク 1,2 の背 景色 G 値 input.bg_color. byte.g - グラフィッ ク 2: 0xCC グラフィック 1,2 の背景色の G 値 の設定 グラフィッ ク 1:未設定 グラフィッ ク 1:未設定 グラフィッ ク 1:未設定 グラフィック 1,2 の背景色の G 値 の設定 グラフィッ ク 1:未設定 グラフィッ ク 1:未設定 グラフィック 1,2 の背景色の B 値 の設定 グラフィッ ク 1:未設 グラフィック 1,2 の背景色の B 値 の設定				ク 1:未設定	
ク 1,2 の背 景色 G 値 byte.g ク 2: 0xCC の設定 グラフィッ ク 1,2 の背 りた ロック input.bg_color. ク グラフィッ ク 1,2 の背 input.bg_color. - グラフィッ ク 1,2 の背 グラフィッ ク 2: 0xCC グラフィック グラフィッ ク 1,2 の背 input.bg_color. - グラフィッ グラフィッ グラフィック ク 2: 0xCC の設定	グラフィッ	input.ba_color	-	グラフィッ	グラフィック 1.2 の背暑色の G 値
景色 G 値 byte.g j 2: 0x000 byte.g グラフィッ ク 1:未設定 グラフィッ ク 1:未設定 グラフィッ ク 1:未設定 グラフィッ ク 1,2 の背 input.bg_color. - グラフィッ ク 1,2 の背 byte.b - 景色 B 値 - グラフィッ ク 2: 0xCC	ク12の背	hyte a		7 2 0xCC	
グラフィッ ク1:未設定 グラフィッ ク1:本設定 グラフィッ ク1:本設定 グラフィッ ク1:本設定	/	byte.g		2.0000	
グラフィッ ク1:未設定 グラフィッ ク1:未設定 グラフィッ ク1,2の背 数te.b シリte.b クラフィック1,2の背景色のB値 ク2: 0xCC				ドニコ・	
グラフィッ input.bg_color. グラフィッ グラフィック 1,2 の背景色の B 値 ク 1,2 の背 byte.b ク 2: 0xCC の設定				リクフノイツ	
グラフィッ input.bg_color. - グラフィッ グラフィック 1,2 の背景色の B 値 ク 1,2 の背 byte.b ク 2: 0xCC の設定 景色 B 値 - - -				ク1:禾設定	
ク 1,2 の背 byte.b ク 2: 0xCC の設定	グラフィッ	input.bg_color.	-	グラフィッ	グラフィック 1,2 の背景色の B 値
景色 B 値	ク 1,2 の背	byte.b		ク2:0xCC	の設定
	景色 B 値				

概略	構造体メンバ	QE for Display [RX]が	サンプルの	設定内容
		出力する define 定義	設定値	
			グラフィッ	
	input heize		ク1:木設正 ガラフィッ	ゲラフィック 1 2 の画像の横幅の
の横幅	Input.nsize	-	クク2:	シリシュンシュンの画家の漫画の設定
			IMAGE_WID	
			TH(480)	
			ガニフィッ	
			ク1:未設定	
画像のデー	input.vsize	-	グラフィッ	グラフィック 1,2 の画像の高さの
タの高さ				設定
			GHT(272)	
			0(22,	
			グラフィッ	
マクロライ	incut offect		ク1:未設定	ゲニファックィックマクロライン
ンオフセッ	input.onset	-	ク2:	クリフィック 1,2 の マンロフィン オフセットの設定
۲ ۲			(IMAGE_WI	
			DTH(480) *	
			グラフィッ	
·>			ク 1:未設定	
クラフィッ ヶ領域の枠	frame edge	-	クラフィッ ク 2 false	グラフィック領域枠を表示/非表示 □ 設定
の表示	llame_eage			
			グラフィッ	
主二明仙冶	·		ク 1:未設定	ダニコノック領域水亚関始位業
□ 衣 示 囲 知 ⊡ 置 x 座 標	coordinate.x	-	クラフィッ ク 2:0	クラフィック限域小十開始1位世 の設定
				- m · · · -
			グラフィッ	
表示開始位	input		ク 1:禾設定 ガニフィッ	ガラフィック領域垂直開始位置
置y座標	coordinate.y	-	ク2:0	の設定
			グラフィッ	
ブリンンド加	bland		ク1:未設定 ガニフィッ	マルファブレンド処理の設定
理の制御設	blend_control	-	ク2:	
定			GLCDC_BL	
			END_CONT	
			ROL_NONE	
			グラフィッ	
			ク 1:未設定	
画像の表示	blend.visible	-	グラフィッ	画像を表示/非表示に設定
設定			ク 2. liue	

	#*たけ えいいぶ		<u></u> பு ⊾, _ீய க	乳白白肉
熌 哈	博道(神メンハ		サノブルの	設定内容
		出力する define 定義	这上他	
			クラフィッ	
			ク1:未設定	
矩形アル	blend.	-	グラフィッ	矩形アルファブレンド領域の枠を
ファブレン	frame_edge		ク2:false	表示/非表示に設定
ド領域の枠				
の表示			グラフィッ	
			ク 1:未設定	
固定アル	blend.fixed	-	グラフィッ	固定アルファ値の設定
ファ値	blend value		ク2:0	(blend control が
	_			GLCDC BLEND CONTROL FIX
			グラフィッ	ED のときのみ有効)
			ク1・未設定	
マルファ値	bland		グースの定	マルコッ値の加減値の恐空
「ルファ直」の加減値	fede apod	-	D D D D D D D D D D D D D D D D D D D	デルノア IEUJIII 減IEUJ設と
の加減値	lade_speed		9 2.0	
			~~	GLCDC_BLEND_
			クラフィッ	
			ク 1:未設定	GLCDC_BLEND_CONTROL_FA
			~~ _	
フレンド処	blend.start_	-	クラフィッ	矩形アルファフレンド領域水平
埋開始位置 	coordinate.x		ク2:0	幅、矩形アルファフレンド水平開
の× 座標				始位置を設定
			グラフィッ	(blend_control が
			ク 1:未設定	GLCDC_BLEND_CONTROL_NO
ブレンド処	blend.end_	-	グラフィッ	NE のときは無効)
理終了位置	coordinate.x		ク2:	
の x 座標			IMAGE_WID	
			TH(480)	
			グラフィッ	
			ク 1:未設定	
ブレンド処	blend.start	-	グラフィッ	矩形アルファブレンド領域垂直
理の開始位	coordinate.v		ク2:0	幅、矩形アルファブレンド垂直開
置の∨座標				始位置を設定
			グラフィッ	(blend control が
			ク1:未設定	GLCDC BLEND CONTROL NO
ブレンド加	blend end		グラフィッ	NEのときは無効)
田の終了位				
一番のヶ底垣	coordinate.y			
			GHT(272)	
			ガニコ・ハ	
			クフノイツ	
<u> </u>			ク □ 木設正	
クロマキー	chromakey.	-		クロマキー処理を有効/無効に設定
処理の有	enbale		ク2: false	
			グラフィッ	
			ク 1:未設定	
クロマキー	chromakey.	-	グラフィッ	クロマキー処理対象 R 値の設定
処理対象 R	before.byte.r		ク 2:0xFF	(chromakey.enbale が false のと
値				きは無効)
			グラフィッ	
			ク 1:未設定	

柙咚	構造体メンバ	OE for Display (BX)が	サンプルの	設定内容
1147.141	構造体パンパ	して Display [パス]が 出力する define 定義	シングルの	成定的各
クロフキー	ahramakay	山刀yるdeline 定我	ガラフィッ	クロフキ――如理対象 C 値の恐空
り ロマイー 処理対象 G	before byte a	-	ク 2: 0vFF	シロマイー処理対象 G illの設定 (chromakov onbalo が falco の k
值	belole.byte.g		2. UNIT	(Chilomakey.enibale が laise のと きけ毎効)
			ガニフィッ	
			クション	
クロマキー	ahramakay		グロスの足	クロフキー処理対象 B 値の設定
ノロマイ 加理対象 B	before byte b	-	ク 2・0xFF	(chromakey enhale が false のと
值	belore.byte.b		2. OAT 1	(chromakey.enbale が laise のと きけ毎効)
			グラフィッ	
			ク1:未設定	
クロマキー	chromakey		グラフィッ	クロマキー処理で置き換えた後の
」 こ、、 置き換え後	after byte a		ク2:0xFF	A 値を設定
A値	anon.byte.a		<i>y</i> <u>2</u> . oki i	(chromakey enhale が false のと
			グラフィッ	(childhaddy).childle 2 haloe ひと きは無効)
			ク 1·未設定	
クロマキー	chromakey		グラフィッ	クロマキー処理で置き換えた後の
プロ 、 、 置き換え後	after byte r		ク2:0xFF	R 値を設定
R 值	anonbyton		2	(chromakey enbale が false のと
			グラフィッ	(emenance) remains in mane ()
			ク1:未設定	
クロマキー	chromakey	-	グラフィッ	クロマキー処理で置き換えた後の
置き換え後	after byte g		ク2:0xFF	G値を設定
G 值	anonbytoig			(chromakey enbale が false のと
			グラフィッ	(きにきには) きにある ひとう こうちょう こうちょう こうちょう こうちょう こうちょう (きにもの)
			ク1:未設定	,
クロマキー	chromakey	-	グラフィッ	クロマキー処理で置き換えた後の
置き換え後	after.bvte.b		ク2:0xFF	B値を設定
B値				(chromakey.enbale が false のと
			グラフィッ	、 きは無効)
			ク 1:未設定	
出力データ	output.endian	-	GLCDC EN	リトルエンディアン/ビッグエン
のビットエ			DIAN_LITTL	ディアンに設定
ンディアン			E	
出力データ	output.	-	RSK RX65N:	出力データのピクセル順序を R-
のピクセル	color_order		GLCDC_CO	G-B 順/ B-G-R 順に設定
順序			LOR_ORDE	
			R_BGR	※ボードによって設定が異なりま
				す。
			RSK RX72N,	
			Envision	
			RX72N,	
			Envision	
			RX65N:	
			GLCDC_CO	
			LOR_ORDE	
			R_RGB	
補正処理の	output.correction_	IMGC_OUTCTL_CALIB_	→	輝度/コントラスト補正の後にガン
美行順序	proc_order	ROUTE		マ補止を実行する。
				または
				カンマ補止の後に輝度/コントラス
				ト補止を実行するように設定

概略	構造体メンバ	QE for Display [RX]が	サンプルの	設定内容
		出力する define 定義	設定値	
ディザ処理	output.dithering.	IMGC_DITHER_ACTIVE	←	True のとき 0 捨 1 入または 2x2
のモード選	dithering_on			パターンディザに設定。
抓				False のとき、切り括てモートに 設定
				(glcdc_cfg_t.output.dithering 以
				トの構造体メンハの設正値は無祝 されます)
ディザ処理	output.dithering.	IMGC_DITHER_MODE	\leftarrow	0 捨 1 入モード/2x2 パターンディ
のモート選 択 2	dithering_mode			サに設定
ディザパ	output.dithering.	IMGC_DITHER_2X2_PA	←	2x2 パターンディザのパターン値
ターン値 A	dithering_			Aの設定(dithering_mode が
	pattern_a			GLCDC_DITHERING_MODE_2X 2PATTERN のときのみ有効)
ディザパ	output.dithering.	IMGC_DITHER_2X2_PB	←	2x2 パターンディザのパターン値
ターン値 B	dithering_			Bの設定(dithering_mode が
	pattern_b			GLCDC_DITHERING_WODE_2A 2PATTERN のときのみ有効)
ディザパ	output.dithering.	IMGC_DITHER_2X2_PC	←	2x2 パターンディザのパターン値
ターン値 С	dithering_			Cの設定(dithering_mode が
	pattern_c			GLCDC_DITHERING_MODE_2X 2PATTERN のときのみ有効)
ディザパ	output.dithering.	IMGC_DITHER_2X2_PD	←	2x2 パターンディザのパターン値
ターン値 D	dithering_			Dの設定(dithering_mode が
	pattern_d			GLCDC_DITHERING_MODE_2X 2PATTERN のときのみ有効)
輝度補正の	output.	IMGC_BRIGHT_OUTCT	←	輝度補正を有効/無効に設定
有効、無効	brightness.	L_ACTIVE		
	enable			glcdc_ctg_t.output.brigntness 以下の構造体メンバの設定値に依ら
				ず、RGB 信号の輝度調整値に0
				が設定されます)
				※QE for Display [RX]では、常に
				true が設定されます。
R 信号の輝	output.	IMGC_BRIGHT_OUTCT		R 信号の輝度調整値の設定
度調整値	brightness.r	L_OFFSET_R		
G 信号の輝 度調整値	output. brightness.g	IMGC_BRIGHT_OUTCT L_OFFSET_G	<i>←</i>	G信号の輝度調整値の設定
B信号の輝	output.	IMGC_BRIGHT_OUTCT	~	B信号の輝度調整値の設定
度調整値	brightness.b	L_OFFSET_B		
コントラス	output.contrast.		→ (コントラスト補正を有効/無効に設
▶1 <t< td=""><td>enable</td><td>CIL_ACTIVE</td><td></td><td> 正 (毎効の埕会</td></t<>	enable	CIL_ACTIVE		正 (毎効の埕会
				alcdc cfg t.output.contrast 以下の
				構造体メンバの設定値に依らず、
				RGB 信号のコントラスト調整値
				に 1.000 が設定されます)
				wge for Display [RA] Cla、常に true が設定されます。

概略	構造体メンバ	QE for Display [RX]が	サンプルの	設定内容
		出力する define 定義	設定値	
R 信号のコ	output.	IMGC_CONTRAST_OUT	<i>←</i>	R 信号のコントラスト調整値を設
ントラスト 調整値	contrast.r	CTL_GAIN_R		定
G 信号のコ	output.	IMGC_CONTRAST_OUT	←	G 信号のコントラスト調整値を設
ントラスト 調整値	contrast.g	CTL_GAIN_G		定
B信号のコ	output.	IMGC_CONTRAST_OUT	\leftarrow	B 信号のコントラスト調整値を設
レトフスト 調整値	contrast.b	CTL_GAIN_B		定
ガンマ補正 の <u>ち</u> 効 毎	output.gamma.	IMGC_GAMMA_ACTIVE	←	ガンマ補正を有効/無効に設定
効	enable			(無効の場合、 alcde efa t.output.gamma 以下の
				構造体メンバの設定値は無視され ます)
				※QE for Display [RX]では、常に true が設定されます。
R 信号のガ	output.gamma.	• gain[16]	<i>←</i>	R 信号の各領域のゲインと開始し
ンマ補正	p_r	IMGC_GAMMA_R_GAIN	(&gs_gamm	きい値の設定
テーフル			a_table_r)	#*生け ノン・ジョーナー ニー ブル 赤粉
		15		「柄垣仲チノハには、テーノル変奴 (as aamma table r)のアドレスを
		Threshold[15]		(90_90
		IMGC_GAMMA_R_TH_0		
		IMGC_GAMIMA_K_TT_T		
G 信号のガ	output.gamma.	• gain[16]	←	G 信号の各領域のゲインと開始し
レマ補正	p_g	IMGC_GAMMA_G_GAIN	(&gs_gamm	きい値の設定
<u>ナ</u> ーノル			a_table_g)	<u> 歩たはょいがにけ テ</u> ーブル亦物
				博坦(PAC) パーは、アークル支索 (gs_gamma_table_g)のアドレス
		Threshold[15]		を設定しています。
		IMGC_GAMMA_G_TH_0		
		11/160_GAN/11/1A_6_111_1		
B信号のガ	output.gamma.	• gain[16]	←	B 信号の各領域のゲインと開始し
レマ補正 ニーブリ	p_b	IMGC_GAMMA_B_GAIN	(&gs_gamm	きい値の設定
		LOU~	a_table_b)	歩告休まいバには テーブル変数
		_15		構造体がシバロな、アージルをか (gs_gamma_table_b)のアドレス
		Threshold[15]		を設定しています。
		IMGC_GAMMA_B_TH_0		
		I~ IMGC GAMMA B TH 1		
		5		
CLUTメモ	clut.enable	-	グラフィッ	CLUT メモリを更新する/しない
リの更新の			ク 2: false	(更新しない場合、clut 以下の構
り インジェン の 選択			ガラフィッ	道体メンハの設定値は無視されま す)
			ク1:未設定	<i>,</i>

QE for Display [RX] サンプルプログラム

概略	構造体メンバ	QE for Display [RX]が 出力する define 定義	サンプルの 設定値	設定内容
CLUT メモ リの先頭ア ドレスへの ポインタ	clut.p_base	-	グラフィッ ク 2: FIT_NO_PT R グラフィッ ク 1:未設定	ポインタが指し示すアドレスから 値を読み出し CLUT メモリにコ ピーします
更新する CLUT メモ リの開始エ ントリ番号	clut.start	-	グラフィッ ク 2: 0 グラフィッ ク 1:未設定	指定したエントリ番号から CLUT メモリの更新を開始します
更新する CLUT メモ リのエント リサイズ	clut.size	-	グラフィッ ク 2: 256 グラフィッ ク 1:未設定	指定したサイズ分の CLUT メモ リを更新します
VPOS 検出 の許可、禁 止	detection.vpos_de tect	-	true	VPOS 検出を許可/禁止に設定
GR1UF 検出 の許可、禁 止	detection.gr1uf_d etect	-	false	GR1UF 検出を許可/禁止に設定
GR2UF 検出 の許可、禁 止	detection.gr2uf_d etect	-	true	GR2UF 検出を許可/禁止に設定
VPOS 割り 込みの許 可、禁止	interrupt. vpos_enable	-	true	VPOS 割り込みを許可/禁止に設定
GR1UF 割り 込みの許 可、禁止	interrupt. gr1uf_enable	-	false	GR1UF 割り込みを許可/禁止に設 定
GR2UF 割り 込みの許 可、禁止	interrupt. gr2uf_enable	-	true	GR2UF 割り込みを許可/禁止に設 定

—:対応する定義なし

←:QE for Display [RX] が出力した定義を設定

7.4 周辺機能

本サンプルで使用する周辺機能を表 7-2 に示します。

機能	用途
グラフィック LCD コントローラ	ディスプレイ表示
(GLCDC)	・グラフィック2を使用
	・グラフィック1は未使用
	・カラーフォーマット RGB565
	・出力データフォーマット RGB565(パラレル 16 ビット)
割り込みコントローラ(ICU)	GLCDC の割り込み制御
拡張 RAM	フレームバッファに使用

表 7-2 使用する周辺機能

7.5 メモリ

本サンプルで使用する ROM/RAM サイズを表 7-3 に示します。

プロジェクト	使用メモリ	サイズ	備考
RSK RX72N	ROM	17266 バイト	コード、定数データ、初期値データ
	RAM	8129 バイト	データ: 376 バイト
			未初期化データ:2633 バイト
			STACK : 5120 バイト
Envision RX72N	ROM	17270 バイト	コード、定数データ、初期値データ
	RAM	8129 バイト	データ:376 バイト
			未初期化データ:2633 バイト
			STACK : 5120 バイト
RSK RX65N	ROM	16842 バイト	コード、定数データ、初期値データ
	RAM	8013 バイト	データ:376 バイト
			未初期化データ:2517 バイト
			STACK : 5120 バイト
Envision RX65N	ROM	16862 バイト	コード、定数データ、初期値データ
	RAM	8013 バイト	データ:376 バイト
			未初期化データ:2517 バイト
			STACK : 5120 バイト

表 7-3 ROM/RAM 使用量

7.6 メモリマップ

本サンプルで使用する固有セクションを表 7-4 に示します。

表 7-4 サンプルで使用する固有セクション

セクション	タイプ	内容
FRAME_BUFFER	data	表示データが格納されるバッファ
		拡張 RAM 領域の固定アドレス(0x00800000)に配置。

7.7 使用割り込み一覧

表 7-5 にサンプルで使用する割り込みを示します。これらの割り込みは GLCDC FIT モジュールで制御され ています。詳細は GLCDC FIT モジュールのマニュアルを参照してください。

表 7-5 サンブルで使	用する割り込み
--------------	---------

割り込み	優先度	説明
VPOS 割り込み	5	グラフィック2ライン検出によって発生する割り込み
		(グループ AL1 割り込み 要因番号:8)
GR1UF 割り込み		バスへのアクセス負荷などにより、グラフィック1アルファブ
		レンド部ヘグラフィックデータの供給が間に合わないときに発
		生する割り込み (グループ AL1 割り込み 要因番号:9)
		本サンプルではグラフィック 1 を使用しないため割り込み禁止
		に設定。
GR2UF 割り込み		バスへのアクセス負荷などにより、グラフィック2アルファブ
		レンド部ヘグラフィックデータの供給が間に合わないときに発
		生する割り込み (グループ AL1 割り込み 要因番号:10)

7.8 本サンプルの使用に際して

本サンプルは、GLCDCを初期化して画面を表示させるために必要な処理を main.c ファイル内に記述して います。ユーザシステムで参考にされる場合には、main.c ファイル内の下記変数や関数と同等のものが必 要です。また、main.c ファイル内で宣言している define 定義は、GLCDC FIT モジュールのパラメータへ代 入する値として使用しているものもありますが、必須ではありません。define 定義の代わりに即値を代入し ても問題ありません。

以下の内容を確認のうえ、ユーザシステムの参考にしてください。

7.8.1 変数一覧

表 7-6 に必要な変数を示します。

型	変数名	内容
static glcdc_cfg_t	gs_glcdc_init_cfg	GLCDC FIT モジュールの R_GLCDC_Open 関 数に渡す構造体変数です。 画面表示に必要な情報を設定します。
static volatile bool	gs_first_interrupt_flag	コールバック関数で初回の VPOS 割り込みを 確認する変数です。
static const gamma_correction_t	gs_gamma_table_r = {gain[16]={IMGC_GAMMA_R_GA IN_00~ IMGC_GAMMA_R_GAIN_15} , Threshold[15]={IMGC_GAMMA_R _TH_01~ IMGC_GAMMA_R_TH_15}}	ガンマ(赤)テーブル テーブルデータは QE for Display [RX]から出力 されるヘッダファイル(r_image_config_(ボー ド名).h)の define 定義を使用しています。
static const gamma_correction_t	gs_gamma_table_g = {gain[16]={IMGC_GAMMA_G_GA IN_00~ IMGC_GAMMA_G_GAIN_15} , Threshold[15]={IMGC_GAMMA_ G_TH_01~ IMGC_GAMMA_G_TH_15}}	ガンマ(緑)テーブル テーブルデータは QE for Display [RX]から出力 されるヘッダファイル(r_image_config_(ボー ド名).h)の define 定義を使用しています。
static const gamma_correction_t	gs_gamma_table_b = {gain[16]={IMGC_GAMMA_B_GAI N_00~ IMGC_GAMMA_B_GAIN_15} , Threshold[15]={IMGC_GAMMA_B _TH_01~ IMGC_GAMMA_B_TH_15}}	ガンマ(青)テーブル テーブルデータは QE for Display [RX]から出力 されるヘッダファイル(r_image_config_(ボー ド名).h)の define 定義を使用しています。

表 7-6 必要な変数

7.8.2 関数一覧

表 7-7 に必要な関数を示します。

表 7-7 必要な関数

関数名	概要
glcdc_initialize	GLCDC の初期化と動作開始の関数です。
qe_for_display_parameter_set	QE for Display [RX]で設定する GLCDC FIT モジュールのパラメータ をまとめた関数です。
glcdc_callback	GLCDC FIT モジュールのコールバック関数です。 GLCDC の割り込みを使用しない場合は必須ではありません。 使用しない場合は、GLCDC FIT モジュールのマニュアルに従い、割 り込みの禁止設定と、コールバック関数へのポインタを無効 (FIT_NO_FUNC)に設定してください。
board_port_setting	ボード依存の端子設定 (LCD パネルのバックライト、リセット制御用 端子)関数です。 使用する LCD パネルや接続に依存するため、必須ではありません。 RSK および Envision では、LCD パネルのバックライトとリセットを 汎用入出カポートで制御する接続になっています。

8. QE for Display [RX]の使用方法

本章では実際のディスプレイ調整の流れに従い QE for Display [RX]の使用方法について説明します。QE for Display [RX]の機能詳細につきましては、QE for Display [RX]に同梱のヘルプを参照ください。

8.1 QE for Display[RX]の起動

e² studio のメニューから『Renesas Views』→『Renesas QE』→『ディスプレイ調整 RX (QE)』を選択 すると QE for Display[RX]が起動します(図 8-1)。

図 8-1 に示すブロック図は、GLCDC の H/W ブロック図を表しており、画像データが出力されるまでの 経路と、各種画質補正が実施される位置関係を知ることができます。また、ブライトネス、コントラストな どの画質調整内容をクリックすることで、画質調整タブに遷移し、各種調整が可能です。



図 8-1 QE for Display[RX]の起動画面

8.2 LCD パネルデータの設定

ユーザが開発するシステムに接続する LCD パネルの情報を設定します。開発システムにディスプレイを 接続するには LCD パネルの仕様とディスプレイコントローラの仕様を摺り合せ、設定可能かつ最適な設定 値を見つけ出す必要があります。設定された情報はその際の比較情報として使用されます。

RSK に搭載の LCD は、Newhaven Display International 社製 NHD-4.3-480272EF-ATXL#-CTP です。また Envision に搭載の LCD は、EastRising 社製 ER-TFT043-3 です。

QE for Display[RX] V1.1.0 のパッケージには、該当の LCD パネルの情報が含まれていますのでそれを選択します。

🖵 ディスプレイ調整 RX (QE) 🔀			🖗 📢 📔 🛱 (Y 💷	▽ □ □
LCD メーカー/種類:	Newhaven Display International, Inc.	~	型名/サイズ:	NHD-4.3-480272EF-ATXL#-CTP	~

LCD パネルの情報設定では表示方式の設定機能があります。表示方式は3つのパターンより選択し設定 します。RSK および Envision 搭載の LCD パネルは表示方式 3(Vsync 信号、Hsync 信号、DE 信号を使用す る方式)を採用しています。

LCD パネルの情報設定方法詳細については、9. LCD パネルデータの設定詳細 を参照してください。

8.3 制御信号の出力設定

QE for Display[RX]の[TCON と LCD 設定]タブを選択し、制御信号の設定を行います(図 8-3)。

このダイアログでは、以下に記載する制御信号の出力設定が可能です。

[パネルドライバ信号(TCON)の出力選択]

出力端子の選択:

LCD_TCON0~LCD_TCON3 端子へ出力(TCON0~TCON3)

制御信号の極性:

正極性(High Active)

負極性(Low Active)

[LCD の設定]

出力データフォーマットの選択:

24bit RGB888 出力 (24bit(GLCDC_OUT_FORMAT_24BITS_RGB888)) 18bit RGB666 出力 (18bit(GLCDC_OUT_FORMAT_18BITS_RGB666)) 16bit RGB565 出力 (16bit(GLCDC_OUT_FORMAT_16BITS_RGB565))

データの出力タイミング:

パネルクロックの立ち上がりで出力 (Rising(GLCDC_SIGNAL_SYNC_EDGE_RISING)) パネルクロックの立ち下がりで出力 (Falling(GLCDC_SIGNAL_SYNC_EDGE_FALLING))

図 8-2 LCD パネル選択

♀ ディスプレイ調整 RX (QE) ※	🗟 ୶ 🔄 🖓 🖗 💷 💎 🗆
LCD メーカー/種類: Newhaven Display International, Inc. V型名/サイズ:	NHD-4.3-480272EF-ATXL#-CTP V
ブロック図から選択「TCONとLCD設定」タイミング設定 画質調整	
「パネルドライバ信号(TCON)の出力選択	
Vsync: TCON0 V High Active V	
Hsync: TCON1 V High Active V	
DE: TCON2 V High Active V	
LCDの設定	
LCD出力フォーマット: 24bit (GLCDC_OUT_FORMAT_24BITS_RGB888) ~	
データ出力のタイミング: Falling (GLCDC_SIGNAL_SYNC_EDGE_FALLING) 〜	

図 8-3 TCON と LCD 設定

各 RSK および Envision での設定はボード仕様より以下となります。

	RSK RX72N / RX65N	Envision RX72N / RX65N
出力端子の選択		
Vsync	TCON0	TCON0
Hsync	TCON2	TCON2
DE	TCON3	TCON3
制御信号の極性	<u> </u>	
Vsync	負極性(Low Active)	負極性(Low Active)
Hsync	負極性(Low Active)	負極性(Low Active)
DE	正極性(High Active)	正極性(High Active)
出カデータフォーマット の選択	16bit RGB565 出力 (16bit(GLCDC_OUT_FORMAT_16BI TS_RGB565)	16bit RGB565 出力 (16bit(GLCDC_OUT_FORMAT_16BI TS_RGB565)
データの出力タイミング	パネルクロックの立ち上がりで出力 (Rising(GLCDC_SIGNAL_SYNC_ED GE_RISING))	パネルクロックの立ち上がりで出力 (Rising(GLCDC_SIGNAL_SYNC_ED GE_RISING))

8.4 LCD パネル制御信号タイミングの調整

デバッガを接続しサンプルプログラムを実行後、図 8-4 に示す設定値を変更することで制御信号タイミン グの変更が可能です。変更した設定値を本ツールより GLCDC のレジスタに直接書き込み、RSK 上の LCD パネルに反映できます。



図 8-4 制御信号タイミングの調整

まずパネルクロック周波数をタイミング調整エリア左上のボックスに入力します。入力したパネルクロック周波数はリフレッシュレートの算出に使用され、画面下部に LCD パネル推奨値との差分として表示されます。パネルクロック周波数はプログラム中で設定する必要があります。QE for Display[RX]から設定することはできません。初期値は LCD パネルの推奨値が設定されています。ここではサンプルプログラム中で設定している 10MHz を入力します。

次に各パラメータを調整します。調整値が赤く表示された場合は、GLCDCの仕様またはLCDパネル仕様 の範囲外の値であることを意味します。この場合、GLCDCの仕様範囲内の値となるよう調整してくださ い。GLCDCの仕様が許容する値の範囲は、赤く表示された調整値へマウスをポイントすることで確認する ことができます。 これらの初期値も LCD パネルの推奨値が使用されます。LCD パネルの水平フロントポーチ(HFP)の推奨 値は2となっており、GLCDC の仕様範囲から外れているため変更が必要です。よって LCD パネル、 GLCDC の仕様を満たす3以上へ変更します。これにより、調整値の表示が赤から黒に変ります。

調整値を決定したところで、本ツール上から GLCDC のレジスタに調整値を書き込み、結果を確認します。

 e^2 × 🔄 🔁 🍳 🛄 🔍 🗖 💭 ディスプレイ調整 RX (QE) 💥 💽 📢 □ (左:設定値をレジスタに設定 右:変更をリアルタイムでレジスタへ設定)

図 8-5 レジスタ設定機能ボタン

レジスタへの反映は、下記の2つの方法があります。

表 8-2 レジスタへの反映機能

ボタン	名前	機能
Ç.	レジスタへ設定	設定値をレジスタに書き込みます。デバッガ接続時のみ有効です。
44	変更をリアルタイ ムでレジスタへ設	オンになっている時、設定値を変更する度にレジスタへ自動で書き 込みます。デフォルトはオフ。
	定	なお、デバッガに接続している場合のみ書き込み処理を行います。 (デバッガ未接続時は何もしません)

注意:

QE for Display[RX]による調整値のレジスタ書き込み機能は、タイミング調整を行うと共にグラフィック 画面をバックグラウンド画面の左上に調整する仕様となっています。

グラフィック画面およびバックグラウンド画面の定義につきましては、『RX65N グループ、RX651 グ ループ ユーザーズマニュアル ハードウェア編』(R01UH0590)または『RX72N グループ ユーザーズマ ニュアル ハードウェア編』(R01UH0824)を参照ください。

本サンプルプログラムでは、青いイメージの外周に赤い 1pixel 幅のラインの描画をします。サンプルプロ グラムでは RSK および Envision に合わせて最適な調整値が設定済みのため調整が不要ですが、実際の開発 では赤い外周のラインが LCD パネルに表示されるように位置合わせを行います。

また、LCD パネルの表示方式および仕様により、各種設定値の細かな変更(数 pixel 単位での移動など)が LCD の表示として現われないことがあります。RSK および Envision 搭載の LCD パネルでは表示方式 3 を 採用しており、バックポーチなどの設定値による表示位置の移動はできません。

8.5 制御信号出力/タイミング調整結果の反映

タイミング調整結果は、ヘッダファイルを出力することでプログラムに反映することができます。QE for Display[RX]のヘッダファイルの生成ボタン(図 8-6)を押下し、設定した制御タイミングを反映したヘッダファイルを生成します。

e ²	– 🗆 X
↓ 「 ディスプレイ調整 RX (QE) ※	🖬 🖬 🔛 😭 🖻 🗖

図 8-6 ヘッダファイルの生成ボタン

「タイミングおよび TCON 設定用ヘッダファイル」のみにチェックをつけ、「生成する」を選択すると、 指定した出力先にヘッダファイルが生成されます。ヘッダファイル名と出力先は、任意で設定可能です。

₽² 生成するヘッダファイル名の指定	×
☑タイミングおよびTCON設定用ヘッダファイル:	
C:¥qe_for_display_rx_sample¥QE_for_Display_sample_RX72N_RSK¥src¥r_lcd_timing_RX72N_RSK.h	参照
□ 画質調整用ヘッダファイル:	
C:¥qe_for_display_rx_sample¥QE_for_Display_sample_RX72N_RSK¥src¥r_image_config_RX72N_RSK.h	参照
生成するキャンセル	ヘルプ(H)

図 8-7 タイミングおよび TCON 設定用ヘッダファイルの生成

本サンプルプロジェクトに反映するには、下記フォルダに r_lcd_timing_< RX72N/RX65N>_< RSK/ Envision >.h の名前で出力し、その後プロジェクトをクリーン、ビルドします。

フォルダ:

<workspace folder> ¥QE_for_Display_sample_< RX72N / RX65N >_< RSK / Envision >¥src

8.6 イメージダウンロード機能

QE for Display[RX]では LCD の特性に合わせて LCD の表示を確認しながら画質を調整します。その際に プログラムを変更せずに LCD に表示するイメージ変更することが可能です。

イメージダウンロード機能では、パソコン上にある LCD に表示するイメージデータ(バイナリファイル)を ダウンロードすることができます。

e ²	_		×
ディスプレイ調整 RX (QE) ※	ା 🐼 ୶ 🔄 😭 🖓 🖸	ļ 🗸 🗆	' 🗖

図 8-8 イメージダウンロード機能ボタン

ツールバーのイメージダウンロード機能ボタンを押します。

e ² イメージを送信		×
送信先アドレス:	0x800000	
サイズ[byte]:		
送信するファイル:	$C: \verb"Yqe_for_display_rx_sample"= QE_for_Display_sample_RX72N_RSK\verb"Yimage"= Load_Sample_480x272_lit.bin"$	参照
	□ 4バイト単位で並びを入れ替えて書き込み	
	送信 キャンセル ヘルフ	/(H)

図 8-9 イメージを送信ダイアログ

表示された[イメージを送信]ダイアログに送信先アドレス、送信するファイルを指定します。[送信先アドレス]のデフォルトはグラフィック2フレームバッファ制御レジスタに設定されている値が設定されます。 本サンプルプログラムでは、グラフィック2を使用していますので変更は不要です。[サイズ]の指定を省略した場合、[送信先アドレス]エディットボックスに指定されたアドレスから[送信するファイル] エディット ボックスに指定されたファイル全体を書き込みます。

本アプリケーションノートには、サンプルのイメージデータが同梱されています。下記を送信します。

フォルダ:

<workspace folder> ¥QE_for_Display_sample_< RX72N / RX65N >_< RSK / Envision >¥image

ファイル:

Load_Sample_480x272_lit.bin

正しく送信が完了すると、図 8-10の様なカラーバーが表示されます。



図 8-10 送信イメージ

8.7 各種画質の調整

[ブロック図から選択]タブの図 8-11 に赤枠で示す画質調整内容をクリックすることで、画質調整タブに 遷移し、各種画質調整が可能です。



図 8-11 各種画質調整ボタンの選択

画質調整タブでは各種画質調整が可能です。QE for Display[RX]では「調整処理順設定」および「ブライトネス」、「コントラスト」、「ガンマ補正」、「ディザ処理」の4つの画質調整機能をサポートします。

これらの調整機能では設定の変更をリアルタイムで反映し、LCDパネルの表示を見ながら調整することができます。

画質調整は簡易設定、またはカスタムにて行います。カスタムで画質調整を行う場合は、『RX65N グ ループ、RX651 グループ ユーザーズマニュアル ハードウェア編』(R01UH0590)または『RX72N グルー プ ユーザーズマニュアル ハードウェア編』(R01UH0824)、および『RX ファミリ グラフィック LCD コ ントローラモジュール Firmware Integration Technology』(R01AN3609)を参照し、各レジスタで設定する 内容と、設定可能な値を確認の上、調整を行ってください。

e ²		– 🗆 X
💭 ディスプレイ調整 RX (QE) 🙁	🛃 📢 📘	😭 e 🗉 🗸 🗖
LCD メーカー/種類: Newhaven Di	isplay International, Inc. 〜 型名/サイズ: NHD-4.3-480272EF-ATX	(L#-CTP ~
ブロック図から選択 TCONとLCD設定	タイミング設定 画質調整	
調整処理順設定 プライトネス コントラスト	х х	
ガンマ補正ディザ処理簡	804 €	
	カスタム: BRTG[9:0] (Green) 804 BRTB[9:0] (Blue) 804 BRTR[9:0] (Red) 804	^

図 8-12 画質調整画面

8.8 画質調整用ヘッダファイルの生成

QE for Display[RX]のヘッダファイルの生成アイコンを押下することで設定した画質調整内容を反映した ヘッダファイルを生成する事が可能です。(図 8-13 参照)

「画質調整用ヘッダファイル」のみにチェックをつけ、「生成する」を選択すると、指定した出力先に ヘッダファイルが生成されます。ヘッダファイル名と出力先は、任意で設定可能です。

e ² 生成するヘッダファイル名の指定	×
□ タイミングおよびTCON設定用ヘッダファイル:	
C:¥qe_for_display_rx_sample¥QE_for_Display_sample_RX72N_RSK¥src¥r_lcd_timing_RX72N_RSK.h	参照
☑ 画質調整用ヘッダファイル:	
C:¥qe_for_display_rx_sample¥QE_for_Display_sample_RX72N_RSK¥src¥r_image_config_RX72N_RSK.h	参照
生成するキャンセル	ヘルプ(H)

図 8-13 画質調整用ヘッダファイルの生成

本サンプルプロジェクトに反映するには、下記フォルダに r_image_config_< RX72N/RX65N>_< RSK/ Envision >.h の名前で出力し、その後プロジェクトをクリーン、ビルドします。

フォルダ:

<workspace folder> ¥QE_for_Display_sample_< RX72N / RX65N >_< RSK / Envision >¥src

9. LCD パネルデータの設定詳細

図 8-1 のダイアログ上部にある[LCD メーカー/種類]のプルダウンリストから[カスタム]を選択するとカス タムディスプレイ・データの編集ダイアログ(図 9-1)が表示されます。このダイアログに LCD パネルの情報 を入力します。

e ² カスタムディスス	プレイ・データの編集				×
√_ カ_ / 種類・	1				~
					*
型名/サイス:					
 ● 表示方式1 ○ 表示方式2 ○ 表示方式3 	CLK IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII				
			空	欄(黄色セル)は入力推	奨欄です
Parameter		Min.	Тур.	Max.	
Panel Clock Fre	equency (PCF) [MHz]	-		-	
Panel Clock Pe	riod (PCP) [ns]	-	-	-	
Horizontal Free	quency (HF) [KHz]	-	-	-	
Horizontal Per	iod (HP) [us]	-	-	-	
Horizontal Tota	al Period (HTP) [Clock]	-	-	-	
Horizontal Pul	se Width (HPW) [Clock]	-		-	
Horizontal Display Period (HDP) [Clock]		-		-	
Horizontal Fro	nt Porch (HFP) [Clock]	-		-	
Horizontal Bac	k Porch (HBP) [Clock]	-		-	
Vertical Freque	ncy (VF) [Hz]	-	-	-	
Vertical Period	(VP) [ms]	-	-	-	
Vertical Total P	eriod (VTP) [Line]	-	-	-	
Vertical Pulse Width (VPW) [Line]		-		-	
Vertical Display Period (VDP) [Line]		-		-	
Vertical Front Porch (VFP) [Line]		-		-	
Vertical Back P	orch (VBP) [Line]	-		-	
URL :					
			OK +t	ツセル ヘルブ	²(H)

図 9-1 カスタムディスプレイ・データの編集ダイアログ

9.1 登録名称の記入

カスタムディスプレイ・データの編集ダイアログ(図 9-2)の『メーカー/種類』、『型名/サイズ』項目に、 任意の名称を入力してください。この名称がドロップダウンリストに登録され、以後選択する事が可能にな ります。

e ² カスタムディスフ	プレイ・データの編集	<
メーカー/種類:	メーカー1 ~]
型名/サイズ:	LCDパネル1]

図 9-2 名称の登録

9.2 表示方式の選択

LCD パネルを接続する為に必要な主な制御信号を表 9-1 主な制御信号 に示します。QE for Display[RX] では、これらの制御信号を組みあわせた3つの方式の表示機器をサポートしています。

名称	機能概略
水平同期信号(Hsync)	表示する1ラインのタイミングを作る信号
垂直同期信号(Vsync)	表示する 1 画面のタイミングを作る信号
パネルクロック(CLK)	表示される画素のサンプリングに使用される信号
表示イネーブル(DE)	有効なデータが出力されていることを示す信号
データ(Data)	表示するデータ

表 9-1 主な制御信号

ユーザは使用する LCD パネルの仕様から、どの制御信号が必要であるかを確認し、表 9-2 使用する制御 信号 に示す 3 つの表示方式から選択する必要があります。

名称	表示方式 1	表示方式 2	表示方式 3	
	СLК	CLK	СLК ···	
水平同期信号(Hsync)	使用	未使用	使用	
垂直同期信号(Vsync)	使用	未使用	使用	
パネルクロック(CLK)	使用	使用	使用	
表示イネーブル(DE)	未使用	使用	使用	
データ(Data)	使用	使用	使用	

表 9-2 使用する制御信号

9.3 制御タイミングの入力

LCD パネルのデータシートより制御タイミングを入力します。Typ.に入力された値がタイミング制御の初 期値に使用され、Min、Max については、QE for Display[RX]のタイミング調整を GUI で行った時、範囲内 であるかどうかの確認に使用されます。

図 9-3 は RSK 搭載の LCD パネルの場合のデータ入力結果です。表 9-3 LCD パネルのデータシートを参照し入力します。

e ² カスタムディスス	プレイ・データの編集			×
メーカー/種類:	メーカー1			~
型名/サイズ:	LCDパネル1			
 ○ 表示方式1 ○ 表示方式2 ⑨ 表示方式3 	CLK IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII			
Parameter		Min.	Тур.	Max.
Panel Clock Fre	equency (PCF) [MHz]	-	9.0	15.0
Panel Clock Pe	eriod (PCP) [ns]	66.666666666	111.111111111	-
Horizontal Free	quency (HF) [KHz]	-	17.14	-
Horizontal Per	iod (HP) [us]	-	58.3430571761	-
Horizontal Tot	al Period (HTP) [Clock]	525	525	605
Horizontal Pul	se Width (HPW) [Clock]	2	41	41
Horizontal Dis	play Period (HDP) [Clock]	480	480	480
Horizontal Front Porch (HFP) [Clock]		2	2	82
Horizontal Bac	k Porch (HBP) [Clock]	2	2	41
Vertical Freque	ncy (VF) [Hz]	-	59.94	-
Vertical Period	(VP) [ms]	-	16.6833500166	-
Vertical Total P	eriod (VTP) [Line]	285	286	399
Vertical Pulse \	Width (VPW) [Line]	1	10	11
Vertical Display	/ Period (VDP) [Line]	272	272	272
Vertical Front Porch (VFP) [Line]		1	2	227
Vertical Back P	orch (VBP) [Line]	1	2	11
URL :				
		OK	キャンセル	へルプ(<u>H</u>)

図 9-3 制御タイミング入力結果

Parameter	Symbol	Spec.			Unit
	Min.	Min.	Тур.	Max.	Onit
Clock cycle	fclk	-	9	15	MHz
Hsync cycle	1/th	-	17.14	-	KHz
Vsync cycle	1/th	-	59.94	-	Hz
Horizontal Signal					
Horizontal cycle	th	525	525	605	CLK
Horizontal display period	thd	480	480	480	CLK
Horizontal front porch	thf	2	2	82	CLK
Horizontal pulse width	thp	2	41	41	CLK
Horizontal back porch	thb	2	2	41	CLK
Vertical Signal					
Vertical cycle	tv	285	286	399	Н
Vertical display period	tvd	272	272	272	Н
Vertical front porch	tvf	1	2	227	Н
Vertical pulse width	tvp	1	10	11	Н
Vertical back porch	rvb	1	2	11	Н

表 9-3 RSK 搭載 LCD パネルのデータシート抜粋

9.4 作成したディスプレイデータの編集

ツールバーのメニューボタンをクリックし、[カスタムディスプレイの編集と削除…]メニューを実行する ことで、作成したディスプレイデータの再編集が行えます。



図 9-4 [カスタムディスプレイの編集と削除…]メニュー

10. ユーザ環境に適用するには

本サンプルをユーザ環境に適用するには、1.1 QE for Display [RX]を使った開発のフローの手順でサンプ ルプログラムを変更する必要があります。本章では、変更点および各手順における補足説明および注意事項 を示します。

以降、RSK RX72N を例として説明します。

ユーザ環境に応じて変更するポイント

<プロジェクトの設定> Smart Configurator の選択(新規プロジェクト作成時) セクションの設定 <SmartConfigurator の設定> クロックの設定 GLCDC FIT モジュールの追加 GLCDC で使用する端子の設定 <プログラムの作成(main.c ファイルの変更)> パネルクロック 端子割り付け(ビットエンディアン) ピクセル配列 その他の制御端子(バックライト、リセットなど) ヘッダファイルの名称 グラフィック2の表示サイズ グラフィック2のベースアドレス

10.1 仕様確認

GLCDC に設定するパラメータの内、QE for Display [RX] でサポートしていないパラメータはユーザが設 定する必要があります。(7.3 GLCDC FIT モジュールのパラメータと QE for Display [RX]が出力するヘッダ との対応を参照。)また、ユーザボードに応じて別途 LCD パネルの制御が必要です。

以下の設定はユーザ環境のボード仕様をもとに決定してください。

- パネルクロック
- 端子割り付け (ビットエンディアン)
- ピクセル配列
- その他の制御端子(バックライト、リセットなど)
- 10.1.1 パネルクロック

GLCDC では、クロックソース(PLL)を分周(1分周~32分周)した周波数をパネルクロックとして使用します。

RSK RX72N に搭載されている LCD パネルの入力クロックは、9MHz(Typ.) ~ 15MHz(Max) です。 本サンプルでは、クロックソースに対して 24 分周しています。

PLL[240MHz](※) ÷ 24 分周 = 10MHz (※) PLL = EXTAL[24MHz] × 10 逓倍 × 1 = 240MHz

/* Output clock */
gs_glcdc_init_cfg.output.clock_div_ratio = GLCDC_PANEL_CLK_DIVISOR_24;

図 10-1 パネルクロックの設定 (クロックソースからの分周比)

結果、本サンプルではパネルクロックを 10MHz で出力しています。

LCD パネルの Typ.が 9MHz に対して 1MHz の差分がありますので、QE for Display [RX]を使用して水平周 波数やリフレッシュレートの調整が必要になってきます。

10.1.2 端子割り付け(ビットエンディアン)とピクセル配列

GLCDC では、マイコンと LCD パネルとの接続に合わせて、端子割り付け(「リトルエンディアン」または「ビッグエンディアン」)とピクセル配列(「RGB」または「BGR」)をそれぞれ選択します。

RSK RX72N に搭載されている LCD パネルとマイコンとの接続は、6.2 端子機能に記載のとおりで、端子割り付け(ビットエンディアン)は「リトルエンディアン」になります。また、ピクセル配列は「RGB」となります。

上記は出力データフォーマットごとに異なります。出力データフォーマットごとの詳細は、対象マイコン のユーザーズマニュアル ハードウェア編を参照ください。

/* Endian */
gs_glcdc_init_cfg.output.endian = GLCDC_ENDIAN_LITTLE;
/* Color order */
gs_glcdc_init_cfg.output.color_order = GLCDC_COLOR_ORDER_RGB;

図 10-2 端子割り付け(ビットエンディアン)とピクセル配列の設定

10.1.3 その他の制御端子

ユーザボードにおける LCD パネルとマイコンの接続に応じて、バックライトやリセットの制御が必要な 場合があります。

RSK RX72N では、6.2 端子機能に記載のとおり、LCD パネルのバックライトやリセットをマイコンの汎 用入出力ポートで制御しています。

static void board_port_setting (void)
{
 /* ---- Port setting ---- */
 /* LCD back light and display-on */
 PORT2.PODR.BIT.B7 = 1; /* Back light */
 PORTK.PODR.BIT.B4 = 1; /* Display */
 PORT2.PDR.BIT.B7 = 1;
 PORTK.PDR.BIT.B4 = 1;

} /* End of function board_port_setting() */



10.2 プロジェクトの作成

10.2.1 Smart Configurator の選択

ユーザが使用するマイコンで新規プロジェクトを作成します。e² studio の画面に従って進めると、図 10-4 の画面が表示されますので、[スマート・コンフィグレータを使用する] にチェックを入れてください。

e – .	×
New Renesas CC-RX Executable Project コーディング・アシストツールの選択	2
✓ スマート・コンフィグレータを使用する □ - ド生成を使用する □ - ド生成を使用する FIT モジュールのダウンロード スマート・コンフィグレータは一つのユーザインタフェースでコード生成と FIT コンフィグレータの機能を併せ持ち、異な プロデバイスドライバーや5ドルウェアをインボート、設定、コードの主成を行わせました。スマート・コンフィグレータの 物をわついのまたまである 思いようまたをである。 レンを使ってあるたちます。 ユンターズロップパイストライ	:るタイ は統一
ドルウェアの使用による周辺機能、割り込み、とン設定のハードウェア酸合を検出します。(スマート・コンフイグル・ サポートデバイスでのみ使用可能です。) User Application Driver and Middleware Driver Code Configured in GUI and Generated MCU Hardware	9G
(ア) (ア) (ア) (ア) (ア) (ア) (ア) (ア) (ア)	JL

図 10-4 コーディング・アシストツールの選択

10.2.2 セクションの設定

フレームバッファとして使用する領域にセクションを設定します。プロジェクトのプロパティを開いて、 FRAME_BUFFER セクションを追加してください。サンプルでは、この FRAME_BUFFER セクションが配 置されている領域に初期画面を生成しています。また、グラフィック2のベースアドレスには、このセクショ ンの先頭アドレスを指定しています。



図 10-5 セクションの設定

なお、本サンプルで使用する拡張 RAM 領域(7.6メモリマップ)は RX72N グループの場合 512K バイトの ため、ユーザ環境で使用する LCD パネルの表示サイズによっては内蔵 ROM への配置を検討する必要があ ります。その際は、プログラムと合わせて画像データも内蔵 ROM にダウンロードしなければならないた め、サンプルの初期画面は使用できません。また、QE for Display [RX]のイメージダウンロード機能も内蔵 ROM へのダウンロードはできません。内蔵 ROM の容量については、対象マイコンのユーザーズマニュア ル ハードウェア編を確認してください。

10.3 Smart Configurator の設定

Smart Configurator を使用してクロックの設定やコンポーネント、端子の設定を行います。全ての設定が 完了したら、コードの生成ボタンをクリックしてコードの生成を行ってください。Smart Configurator の詳 細は Smart Configurator のマニュアルを参照してください。

10.3.1 クロックの設定

Smart Configurator の[クロック]タブより、ユーザ環境に合わせてクロックの設定を行ってください。また、PLL はパネルクロックのクロックソースになりますので周波数を間違えないようご注意ください。

10.3.2 GLCDC FIT モジュールの追加

Smart Configurator の[コンポーネント]タブより、GLCDC FIT モジュールをプロジェクトに追加してくだ さい。コンポーネントの追加ボタンをクリックし、[コンポーネントの追加]ダイアログのコンポーネントー 覧から GLCDC FIT モジュール(r_glcdc_rx)を選択します。モジュールのダウンロードがまだの場合は、 このダイアログの[他のソフトウェアコンポーネントをダウンロードする]からダウンロードできます。

10.3.3 GLCDC で使用する端子の設定

Smart Configurator の[コンポーネント]タブより、追加された GLCD FIT モジュール(r_glcdc_rx)を選択 すると、GLCDC で使用可能な端子の一覧が表示されます。ユーザ環境に合わせて各端子を設定してください。

[コンポーネント]タブの入力後、[端子]タブで各端子機能に割り当てるポート番号を指定してください。 RSK RX72N のサンプルでは、7.2.3 GLCDC の端子設定のとおりに設定しています。

10.4 QE for Display [RX]による調整(初期設定)

8.QE for Display [RX]の使用方法と 9. LCD パネルデータの設定詳細の内容をもとに、QE for Display [RX] を使って、ユーザ環境で使用する LCD パネルの情報を入力し、最初の調整を行います。

調整の際は、GLCDC FIT モジュールの設定範囲に注意してください。QE for Display [RX]の[タイミング 設定]タブにおける各設定値は、登録したパネルデータおよび GLCDC の設定範囲に違反するすると赤く表 示されます。そのため、黒く表示されている状態であっても、GLCDC FIT モジュールの設定範囲に違反し ている場合があります。

調整が完了したらヘッダファイルを生成します。

RSK RX72N のサンプルでは、QE_for_Display_sample_RX72N_RSK¥src フォルダ直下に

r_lcd_timing_RX72N_RSK.h

r_image_config_RX72N_RSK.h

の名称でヘッダファイルを生成しています。

10.5 プログラムの作成

本サンプルをベースにプログラムを作成します。

10.5.1 サンプルプログラムのコピー

プロジェクト生成時に作成された「新規プロジェクト名」.c ファイルを削除し、本サンプルの main.c ファイルを新規作成したプロジェクトにコピーします。

10.5.2 プログラムの変更

7.8 本サンプルの使用に際して や 10.1 仕様確認で確認した仕様を参考に main.c ファイルを変更し、プログラムを完成させます。このとき、下記の内容について変更が必要な場合があります。

(a) ヘッダファイルの名称

本サンプルの main.c ファイルでインクルードしているヘッダファイルを QE for Display [RX]で生成した ヘッダファイルに変更してください。

/* Header files for RSKRX72N board output by QE for Display [RX] */
#include "r_image_config_RX72N_RSK.h"
#include "r_lcd_timing_RX72N_RSK.h"

図 10-6 ヘッダファイルのインクルード

(b) グラフィック2の表示サイズ

使用する LCD パネルの表示サイズに合わせて define 定義の値を変更してください。この define 定義を本 サンプルで表示される初期画面のデータ生成にも使用しているため、定義値の変更に合わせて初期画面の表 示サイズも変更されます。(main.c の frame_buffer_initialize 関数)

/* Image info definition */
#define IMAGE_WIDTH (480u) /* Width of image used in this sample. */
#define IMAGE_HEIGHT (272u) /* Height of image used in this sample. */
/* ---- Graphic 2 setting ---- */
/* Image format */
gs_glcdc_init_cfg.input[GLCDC_FRAME_LAYER_2].hsize = IMAGE_WIDTH;
gs_glcdc_init_cfg.input[GLCDC_FRAME_LAYER_2].vsize = IMAGE_HEIGHT;

図 10-7 グラフィック2の画像サイズ指定

(c) グラフィック2のベースアドレス

FRAME_BUFFER セクションの領域を使用しない、ROM に配置したデータを表示したいなどの場合は、 グラフィック2のベースアドレスを、画像データが格納されている領域の先頭アドレスに変更してください。

```
/* ---- Graphic 2 setting ---- */
/* Image format */
gs_glcdc_init_cfg.input[GLCDC_FRAME_LAYER_2].p_base = (uint32_t *) FRAME_BUF_BASE_ADDR;
```

図 10-8 グラフィック2のベースアドレス指定

10.6 実行から調整終了まで

プログラムの作成が完了したらデバッガを起動してプログラムを実行してください。初期画面が正常に表示されない場合、正しく設定されていません。QE for Display [RX] による調整値、および GLCDC FIT モジュールのパラメータ設定などを確認してください。

図 10-9 にトラブルシューティングを示します。



図 10-9 トラブルシューティング

改訂記録

		改訂内容		
Rev.	発行日	ページ	ポイント	
1.00	Jan.31.18	-	初版発行	
1.10	Apr.07.20	-	RSK RX72N, Envision Kit RX72N のサポート追加	
		11	「図 4-1 プロジェクトのフォルダ構成」変更	
		14-19	図を e² studio v7.7.0 と RSK RX72N 用プロジェクトを使	
		38-52	用した画面に変更	
		26	表 7-1「設定内容」欄にある、一部説明を変更	
		34	「表 7-3 ROM/RAM 使用量」を更新	

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテク ニカルアップデートを参照してください。

1. 静電気対策

CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。

2. 電源投入時の処置

電源投入時は、製品の状態は不定です。電源投入時には、LSIの内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リ セット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。同様に、内蔵パワーオンリ セット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

3. 電源オフ時における入力信号

当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れないでください。入力信号や入出力プルアップ電源からの電流注入に より、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」について の記載のある製品は、その内容を守ってください。

4. 未使用端子の処理

未使用端子は、「未使用端子の処理」に従って処理してください。CMOS 製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI 周辺のノイズが印加され、LSI 内部で貫通電流が流れたり、入力信号と認識 されて誤動作を起こす恐れがあります。

5. クロックについて

リセット時は、クロックが安定した後、リセットを解除してください。プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した 後に切り替えてください。リセット時、外部発振子(または外部発振回路)を用いたクロックで動作を開始するシステムでは、クロックが十分安定 した後、リセットを解除してください。また、プログラムの途中で外部発振子(または外部発振回路)を用いたクロックに切り替える場合は、切り 替え先のクロックが十分安定してから切り替えてください。

6. 入力端子の印加波形

入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、V_L(Max.)から V_H(Min.)までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、V_L(Max.)から V_H (Min.)までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。

リザーブアドレス(予約領域)のアクセス禁止
 リザーブアドレス(予約領域)のアクセスを禁止します。アドレス領域には、将来の拡張機能用に割り付けられている リザーブアドレス(予約領域)があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

8. 製品間の相違について

型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。同じグループのマイコンでも型名が違うと、フラッシュメモリ、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ幅射量などが異なる場合が あります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

ご注意書き

- 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に 起因して生じた損害(お客様または第三者いずれに生じた損害も含みます。以下同じです。)に関し、当社は、一切その責任を負いません。
- 当社製品、本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、 著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではあ りません。
- 3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
- 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改 変、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
- 5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図 しております。

標準水準: コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット等 高品質水準:輸送機器(自動車、電車、船舶等)、交通制御(信号)、大規模通信機器、金融端末基幹システム、各種安全制御装置等 当社製品は、データシート等により高信頼性、Harsh environment向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある 機器・システム(生命維持装置、人体に埋め込み使用するもの等)、もしくは多大な物的損害を発生させるおそれのある機器・システム(宇宙機器 と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等)に使用されることを意図しておらず、これらの 用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責 任を負いません。

- 6. 当社製品をご使用の際は、最新の製品情報(データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等)をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
- 7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする 場合があります。また、当社製品は、データシート等において高信頼性、Harsh environment向け製品と定義しているものを除き、耐放射線設計を 行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客 様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を 行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行って ください。
- 8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用 を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことに より生じた損害に関して、当社は、一切その責任を負いません。
- 9. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
- 10. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものといたします。
- 11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
- 12. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。
- 注 1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社が直接的、間接的に 支配する会社をいいます。
- 注2. 本資料において使用されている「当社製品」とは、注1において定義された当社の開発、製造製品をいいます。

(Rev.4.0-1 2017.11)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24 (豊洲フォレシア) www.renesas.com

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の 商標です。すべての商標および登録商標は、それぞれの所有者に帰属 します。

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓 ロに関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/